太空船泰坦尼克号：预测哪些乘客被传送到另一个维度

赵建桥[1]，徐梓浩[1]

[1](南京信息工程大学 计算机学院,江苏 南京 210044)

摘　要：本文尝试使用基本的缺失值填充和数值转换的方法对数据集进行预测效果不好后（1921/2637），对数据进行探索性分析，大幅优化特征工程和缺失值处理方法，最后得到了 8%的好成绩（234/2637）。本文采用了一系列综合性的数据预处理和特征工程方法,其中最突出的是对于不同的类型数据进行特征探索和联合分布分析填补缺失值的方法。本文对于连续性数据进行区间分段分析，对于字符型数据采用分字段分析法，结合了很多人文科学的方法。缺失值处理方面，与传统的均值填充或众数填充法不同，联合分布分析方法可以更精准地填补缺失值，因为它考虑了数据的多个属性特征之间的联合分布规律。此外，对于数据预处理，本文采取了如去除影响较小的类别特征变量、采用 onehot 编码和取对数处理等方法，进一步提高了模型的准确性和稳定性。最后，在建模方面，本文还综合使用了十余种分类器模型，并通过集成学习投票和 stack 堆叠法的方式对模型进行优化。最后的结果上传至 kaggle 进行审核，准确率为 0.80827，排名 8%（234/2637）。

关键词：联合分布；特征工程；onehot 编码；集成学习投票，stack 堆叠法

Spaceship Titanic: Predict which passengers are transported to an alternate dimension

Abstract: This article tries to use basic missing value filling and numerical conversion methods to predict the data set with poor results (1921/2637), then conducts exploratory analysis on the data, greatly optimizes feature engineering and missing value processing methods, and finally gets 8% Good results (234/2637). This paper adopts a series of comprehensive data preprocessing and feature engineering methods, the most prominent of which is the method of feature exploration and joint distribution analysis for different types of data to fill in missing values. In this paper, the continuous data is analyzed by interval and segment, and the character data is analyzed by field, which combines many methods of humanities. In terms of missing value processing, unlike the traditional mean filling or mode filling methods, the joint distribution analysis method can more accurately fill in missing values because it takes into account the joint distribution of multiple attribute features of the data. In addition, for data preprocessing, this paper adopts methods such as removing category feature variables with less influence, using onehot encoding and logarithmic processing, which further improves the accuracy and stability of the model. Finally, in terms of modeling, this paper also comprehensively uses more than ten classifier models, and optimizes the model by means of ensemble learning voting and stack stacking. The final result was uploaded to Kaggle for review, with an accuracy rate of 0.80827 and a ranking of 8% (234/2637).

Key words: joint distribution; feature engineering; onehot encoding;ensemble learning voting

目录

基本方法

太空船泰坦尼克号：预测哪些乘客被传送到另一个维度(Spaceship Titanic：Predict which passengers are transported to an alternate dimension)这个任务，是一个二分类预测的问题。对于这个问题，本文将采取以下的方法和步骤：

1. 实验数据描述。
   这一步中，本文将对这次任务的数据集进行初步的分析，探究出特征参数及每个特征参数的含义，初步分析标签（待预测项）的含义。同时，针对本题的训练集和测试集大小做初步探究，确定研究的方法和数据量。
2. 缺失值和数据类型分析。
   对于乘客这个 10000 多个样本的数据集，含有缺失值是必然的。在这一步中，本文将探索缺失值的**占比以及分布情况**，并针对每一个特征参数进行单独分析，包括他们的数据类型，以及唯一值数量，并将这些属性分类为连续型的数值型属性，描述型属性和类别特征属性。
3. 模型建立的初步尝试
   本文先使用最基本的方法初步建立模型测试结果。对于连续性变量采用均值填补法，对于非连续性变量采取众数填补法。将非数值型变量采用 LabelEncoder 类的 fit_transform()方法进行转化，最后使用多个分类器进行训练分析。
4. 模型的改进
   接下来就是对迷信的改进分析。
5. 探索性数据分析。
   根据对于数据类型的分析结果，这一步将分别对每一种属性进行探索。对于连续型变量，本文将其用**区间划分**的方式进行分析，探索他们和标签之间的关系。对于类别特征变量，本文主要探索他们和标签之间的**关联度**，关联度低的，在后续降维的环节中可以去除。而对于描述型变量，本文对他们进行**字段拆分分析**，尤其是 PassengerID 和 Cabin，他们中包含了不止一个字段，将其拆开分析能更好的探究他们对分类结果的影响。
6. 特征工程。
   这一步主要对数据按照上一步数据分析的结果进行处理。对于连续的数值型属性，对他们采用区间划分的方式进行分组。对于描述型变量，对他们的每一个字段进行统计学分析，并通过现实意义将他们分组，比如**乘客的姓氏相同的可以认为是一个家族**，将所有的**消费属性归为一类消费**，抑或是**根据 ID 来对乘客组别大小进行特征提取**等等。
7. 处理缺失值。
   这一步并没有直接采用传统的均值填充或众数填充法，而是综合上述特征工程和数据分析结果，采用联合分布分析的办法，逐级的将缺失值进行填补。数据分析和特征工程中，很多潜在的规律被挖掘出来，比如**每一组的乘客都来自于同一星球**，因此可以采用这个规律最大精度地填补星球这一 HomePlanet，而传统的统计学方法则没有这么高的精度。
8. 数据预处理。
   这一步主要将特征工程中分类过的母数据进行去除，并**去除影响很小的类别特征变量**。另外，对分布差距很大的数值型变量做取**对数处理**，使他们的分布更加均匀，并对类别特征属性采用 **onehot 编码**。
9. 主成分分析。
   做出方差累计曲线，分析出应该取多少主成分算作合理。
10. 模型建立和提交。
    探索 10 余种不同分类器之间的性能差距。并采用集成学习，通过**投票**的方式尝试优化模型的准确性。并且，针对效果特别好的模型，对其进行单独**调参**分析。最后，将最优的模型提交 kaggle 官方。

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：

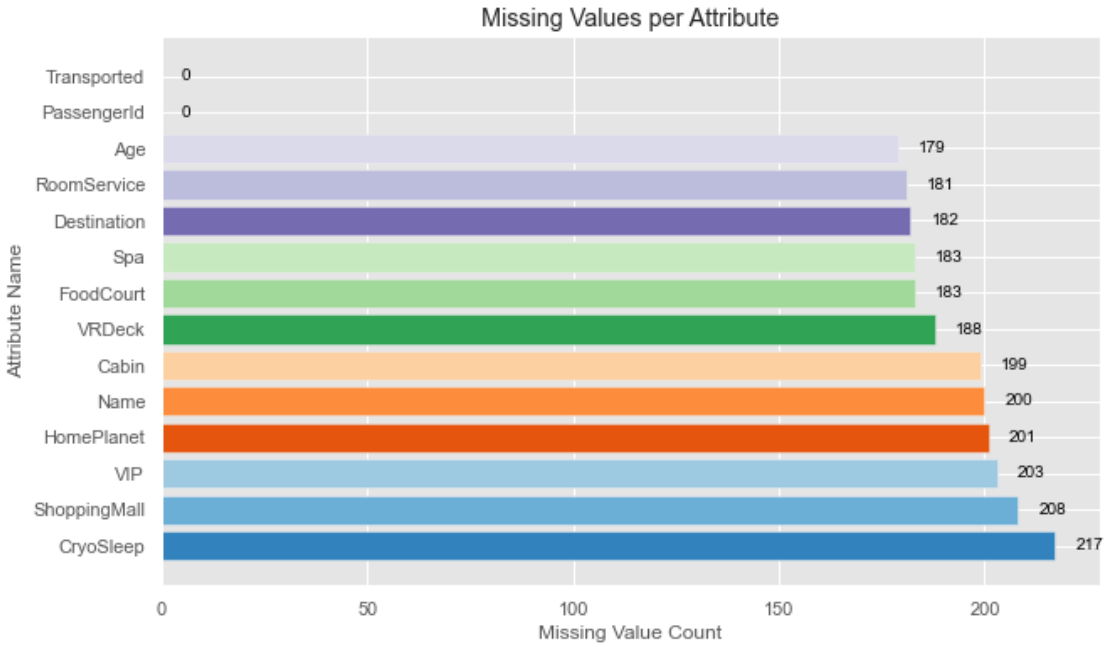

图 1 各属性缺失值数量

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：

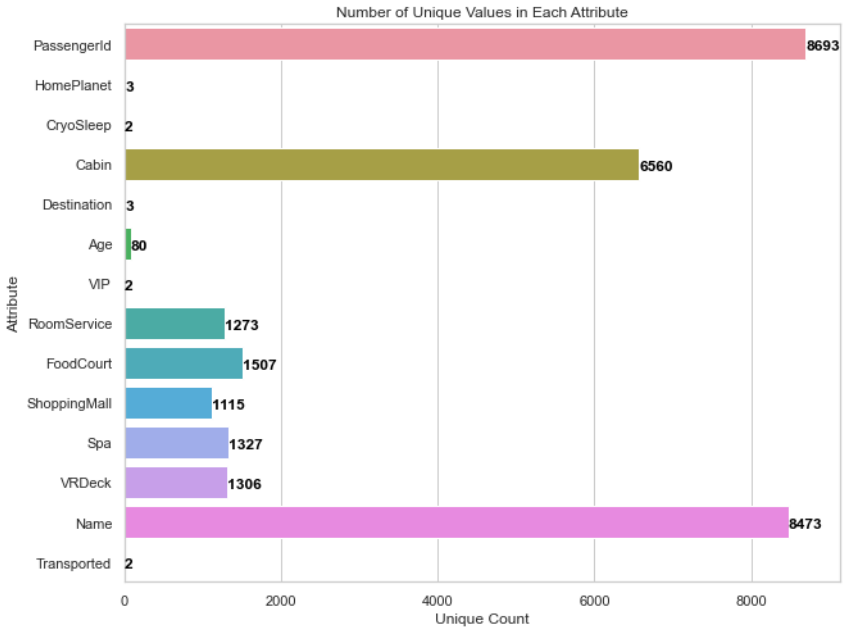

图 2 各属性的唯一值数量

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
$\boldsymbol{fit\_transform()}$	Destination, HomePlanet, Cabin
$(\boldsymbol{true}, \boldsymbol{false}) \rightarrow (\mathbf{1}, \mathbf{0})$	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：

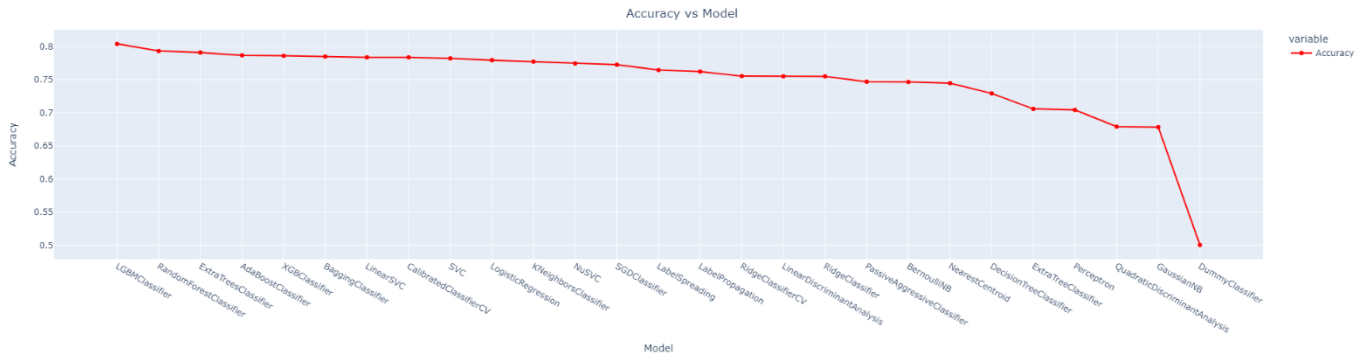

图 3 各种分类器结果对比

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。

Catboost 算法对于高维度数据和异常值的处理很不错，于是本文也尝试使用 catboost 的算法来尝试这个数据集的预测工作。得到的结果如下表所示：

Catboost	Accuracy	Auc	F1-score
结果	0.80	0.80	0.80

表 4 catboost 的结果

对于 lazypredict 模型中分析可得，最好的模型是 LGBMClassifier，最后尝试使用 lazypredict，LGBMClassifier 和 RandomForest 模型投递至 kaggle 官方，测试未知数据集的准确度，得到的结果并不理想，public score 仅有 0.78396，在排名上仅是 75%，显然不令人满意。所以这个模型亟待优化。

反思和模型的改进

模型的问题

对于上述的模型，有几个显著的问题。

a) 对于缺失值的填补太过粗糙，对于连续型的变量没有做一定的分类，粗暴地使用均值填充，会导致很多特征丢失，对最后的精度产生影响。

b) 对于 object 型的变量，其中很多属性可以拆分字段进行分析。比如 name 不能粗糙地直接丢弃，根据常识可以知道同一个家族中的乘客的姓是一样的，这其中可能存在着一些待发现的特征。Cabin 字段也可以拆分出来进行进一步分析。

c) 对于本地的测试集的精确度有 80，但是投递到 kaggle 仅有 78，说明丢失了很多隐藏在数据中的特征，也或者是模型过拟合了。

d) 乘客的信息分布每一个属性并不是孤立的，可以使用联合分布的方法对其进行分析达到最好的效果。

e) 对于效果比较好的几个模型，可以采用投票法进行优化。也可以对于单独的一个模型进行细微的调参，以达到最好的效果。

因此，本文绝对在数据分析，缺失值填充，特征工程，数据预处理部分进行进一步精进。

探索性数据分析

为什么最后的结果不尽如人意，最根本的原因是对数据本身的特性没有分析。探索性数据分析是必不可少的。

由于这次实验的目的是需要预测乘客是否被传送到另外一个星系，即 transported 的取值，所以对于 transported 的分布有一定的要求，如果数据分布的不平均还需要采取上采样或者下采样的策略。如下图是对 transported 的分布规律的探索：

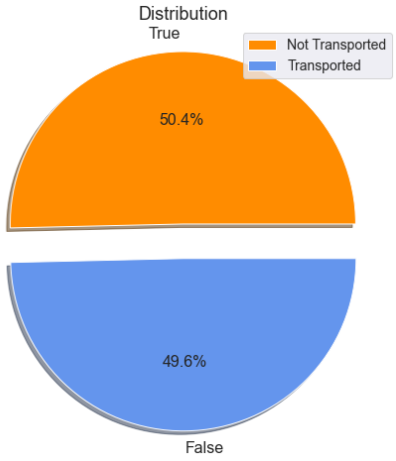

图 4 transported 的分布

可以发现，transported 基本是对半分的，所以这个数据集是比较均衡的，不需要进行上采样或者下采样的处理。

接下来对于各个属性进行单独分析。

连续性变量

首先是年龄。由第一步分数据分析可知，年龄的不同值数量仅有 80 中不同的取值且均为整数，所以可以使用直方图图来分析连续变量年龄的分布情况。直方图如下所示：

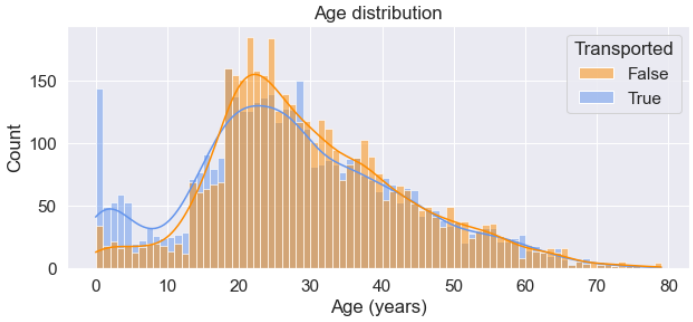

图 5 年龄直方图

从中可以得出以下结论：

a) 0-18 岁的乘客中，传送成功的比例更高

b) 18-25 岁的乘客中，传送成功的比例更低

c) 25 岁以上的乘客中，传送成功的和不成功的比例基本相同

这给到我们一个启示可以按照三个年龄段对于数据进行划分，这样可以更好的进行模型的训练。

接下来是对各个付费服务属性（RoomService，FoodCourt，ShoppingMall，Spa，VRdeck）的分析。由于各个花费是连续性的变量，所以仍然可以使用直方图的形式行进分析。以下是直方图：

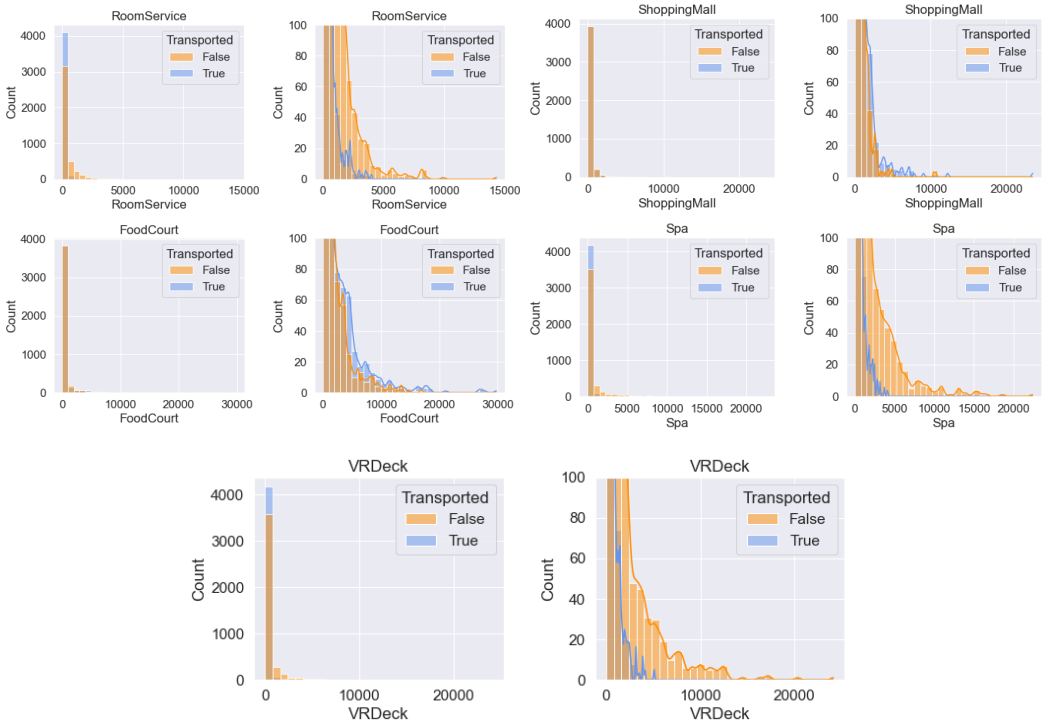

图 6 付费服务属性直方图

从中我们可以得到下面的结论：

a) 大多数的人没有花多少钱

b) 花费的分布呈指数衰减（如图 3 所示）

c) 存在一小部分异常值

d) 被传送的人倾向于花费较少

RoomService（客房服务）、Spa（温泉）和 VRDeck（虚拟现实甲板）与 FoodCourt（美食广场）和 ShoppingMall（购物中心）有不同的分布 - 我们可以将其视为奢侈品与基本设施。

这些在数据处理当中可以给我们下面的启示：

a) 创建一个新的特征，跟踪所有 5 个设施的总支出。

b) 创建一个二进制特征，用于指示该人是否没有花费任何金额（即总支出为 0）。

c) 进行对数转换以减小偏度。

类别特征变量

经第一步分析得，'HomePlanet', 'CryoSleep', 'Destination', 'VIP'，这些属性属于类别特征变量。接着用可视化的方法，生成每个分类特征的计数图，其中 x 轴表示特征中的类别，y 轴表示每个类别中的观测计数。这些计数图如下所示：

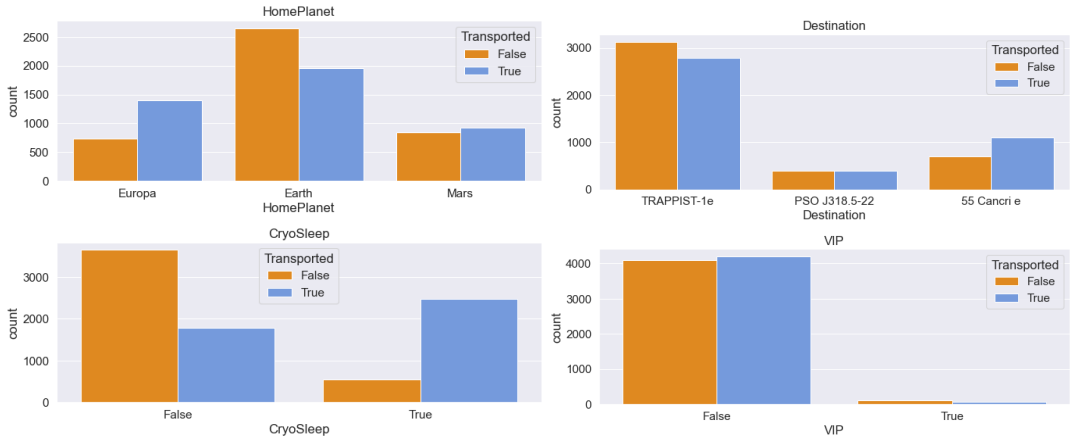

图 7 各个分类计数图

从中可以得到下面的结论：

a) VIP 这个信息由于在传送成功和不成功的乘客里面基本都是对半分布，因此这个特征对于我们的模型训练没有什么帮助，所以我们可以将其删除

b) CryoSleep 很重要，因为在传送成功的乘客中，大多数都是在冷冻睡眠中醒来的

因此可以将 VIP 特征删除。

定性特征的分析

首先使用 head()函数显示前五个记录，初步分析数据的格式。

	PassengerId	Cabin	Name
0	0001_01	B/0/P	Maham Ofracculy
1	0002_01	F/0/S	Juanna Vines
2	0003_01	A/0/S	Altark Susent
3	0003_02	A/0/S	Solam Susent
4	0004_01	F/1/S	Willy Santantines

表 5 定性特征前 5 条记录

从中可以得出：

a) PassengerId 的形式为 gggg_pp，其中 gggg 表示乘客所在的组，pp 表示该组中的编号。

b) Cabin 的形式为 deck/num/side，其中 side 可以是 P 表示舷侧（Port），或者是 S 表示舷侧（Starboard）。

因此，

a) 可以从 PassengerId 特征中提取组别和组内人数。

b) 可以从 Cabin 特征中提取甲板（deck）、编号（number）和舷侧（side）。

c) 可以从姓名（Name）特征中提取姓氏以识别家庭。

总之，根据上述分析，现在可以做特征工程。

特征工程

这一步是根据数据探索中获得的方向对于数据进行处理。

根据上文对年龄的分析，可以对年龄进行分组，分组结果如下：

Age_group	Total	Transported	Percent_Transported
Age_0-12	806	564	69.97519
Age_13-17	739	409	55.34506
Age_18-25	2351	1077	45.81029
Age_26-30	1207	599	49.62717
Age_31-50	2674	1282	47.94316
Age_51+	737	357	48.43962

表 6 年龄分组结果

不同组内的传送的成功率有差距，基本符合本文之前数据探索当中得出的结论。

由于 RoomService，FoodCourt，ShoppingMall，Spa，VRdeck 这些属性都是关于各个乘客的消费数量，所以本文将它合并为一类进行分析，即消费类。

首先计算总的消费并定位没有消费的乘客。结果直方图如下所示：

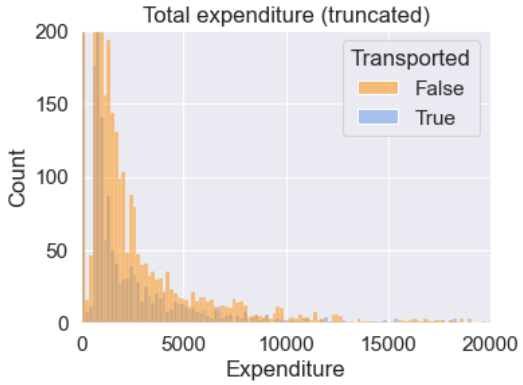

图 8 消费直方图

从数据的分布可以看书，大部分乘客的消费集中在 0 到 5000，极少部分乘客的消费超过 5000。

接下来，将乘客分为消费和未消费两类，结果如下表所示。

	Total	Transported	Percent_Transported
0（未消费）	5040	1505	29.86111
1（已消费）	3653	2873	78.64769

表 7 消费分类表

其中可以看到消费为 0 的乘客中，传送成功的比例要高于传送失败的比例，这可以作为一个特征。

接下来是对乘客 ID 进行分析。使用直方图分析如下：

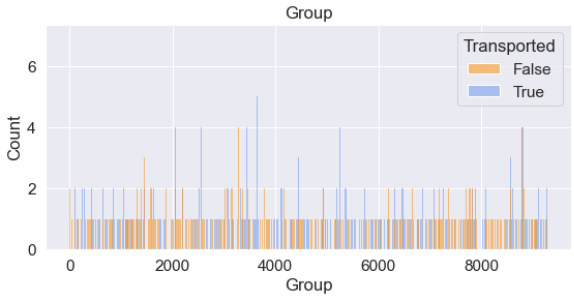

图 9 乘客分组直方图

可以发现分组过多，基数太大（6217），使用 one-hot 会导致维度爆炸。另一方面，组的大小是一个有用的特征，因为组的人数仅有 1-8 这 8 中情况。于是根据组的大小的分类表结果如下所示：

Group_size	Total	Transported	Transported(100%)
1	4805	2174	45.24454
2	1682	905	53.80499
3	1020	605	59.31373
4	412	264	64.07767
5	265	157	59.24528
6	174	107	61.49425
7	231	125	54.11255
8	104	41	39.42308

表 8 分组大小表

可以发现，单独出行的人被传送的概率小于组队出行的，因此可以作为一个特征，即乘客是否单独出行。这个特征做出的直方图如下所示：

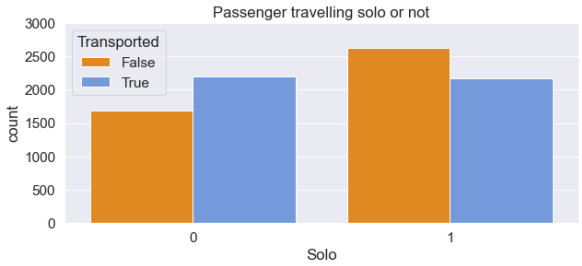

图 10 乘客是否单独出行直方图

所以，乘客是否单独出行可以作为一个特征。

接下来是对 Cabin 信息的解析。

从上述的分析可以得知，Cabin 分为三个字段，第一个字段是甲板的信息，第二个字段是编号，第三个字段是甲板的方位。分别对三个字段进行解析。做出的直方图如下所示：

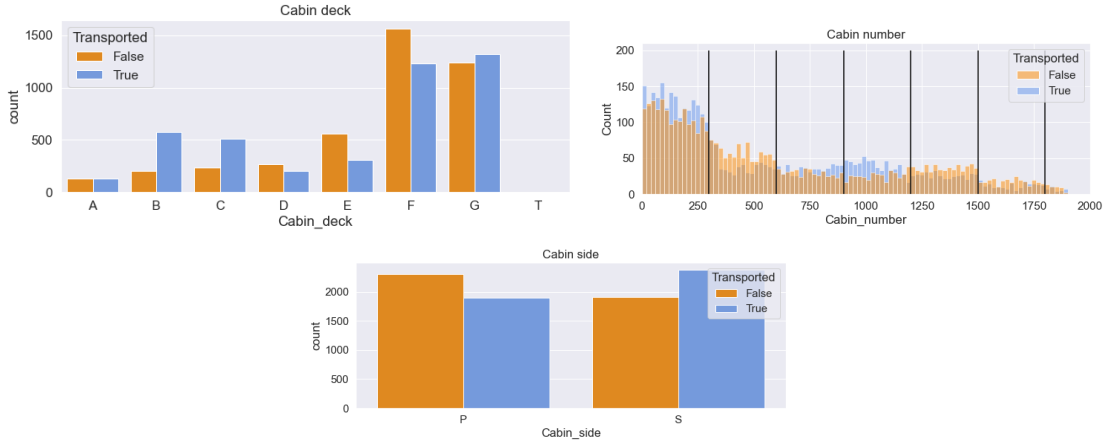

图 11 Cabin 三个字段直方图

对于 Cabin number 分组可知，被分组成了每组 300 个舱室。这意味着我们可以将这个特征压缩成一个分类特征，指示每位乘客所在的舱室组。

Cabin_number 分组结果如下图所示：

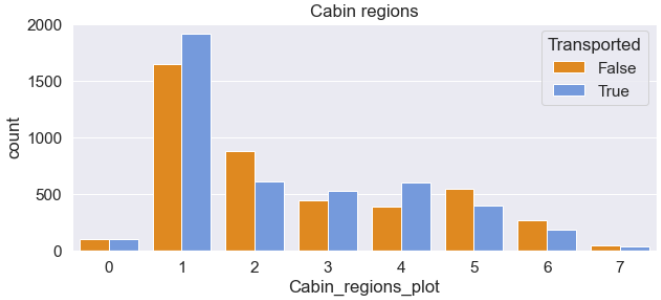

图 12 Cabin_number 分组结果

因此，可以将 Cabin_number 作为一个分类特征。

另外，各位乘客的姓氏可以作为一个家族的标志，根据家族也可以形成一个分类特征。

这里，对各个家族的人数进行统计，并把不同人数的家族的个数进行统计，结果如下表所示：

Family_size	Transported	Not Transported
1	83	50
2	228	197
3	341	275
4	425	380
5	521	458
……	……	……

表 9 家族人数统计表（仅展示 1-5）

同样地，家族成员数量也可以作为一个特征。

处理缺失值

缺失值探索

在处理缺失值的时候，合并训练集和测试集来处理缺失值，两个放在一起处理缺失值更有利于从总体上把握特征，能够更好的处理缺失值。当然在完成缺失值处理之后还是要将训练集和测试集分开的。

首先，先对缺失值的分布进行探索，对于缺失值的分布和比例如下表所示：

	Number_missing	Percentage_missing
HomePlanet	288	2.22
CryoSleep	310	2.39
Destination	274	2.11
Age	270	2.08
……	……	……

表 10 缺失值分布表（仅展示 4 个属性）

由于所有的特征的缺失数量均在 200-400 之间，且百分比都在 2%-3%之间，所以这里仅展示四个特征。接下来，使用热力图对缺失值进行进一步探索：

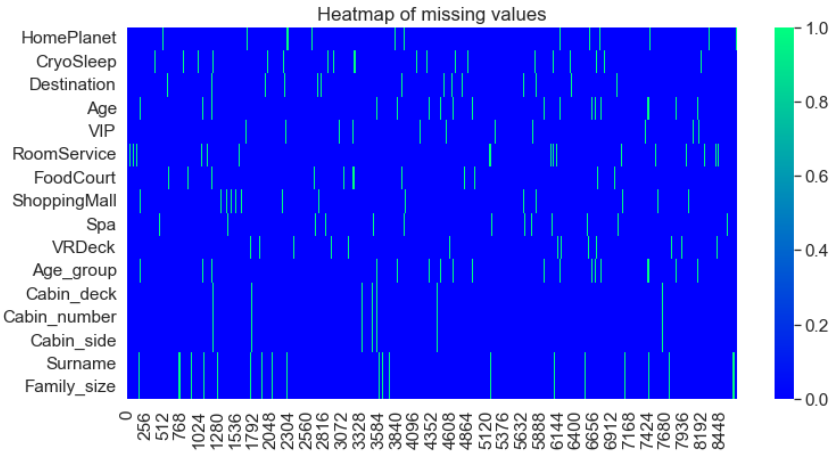

图 12 缺失值热力图

从总体上看来缺失值占比大概在 2%-3%，这个比例不算大。但是这可以看出，缺失值的分布并不集中，也就是说，缺失一个参数的样本数量可能会很大，所以要进一步分析。

接下来是根据缺失值的个数对样本进行统计，结果如下表所示：

Number of missing values	Count	Percentage
0	6606	75.99
1	1400	16.10
2	421	4.84
3	217	2.50
4	40	0.46
5	6	0.07
6	3	0.03

表 11 缺失值数量统计

可以看到大部分的记录缺失都是缺 1 个数据，有极少数的情况出现了缺了 2-3 个属性的，缺失 3 个以上属性的占比很小。

可以从中得出以下的结论：

a) 缺失值与目标变量独立，大部分情况下是孤立存在的。

b) 尽管数据中只有 2%的缺失值，但大约 25%的乘客至少有一个缺失值。

c) PassengerId 是唯一一个没有任何缺失值的（原始）特征。

a) 所以，这些结论对填补缺失值有以下的帮助：

d) 由于大部分缺失值是孤立存在的，与其删除行，填充这些缺失值是有意义的。

e) 如果 PassengerId 与其他特征之间存在关联，我们可以根据该列填充缺失值。

处理缺失值的最简单方法是对连续特征使用中位数，对分类特征使用众数。这种方法足够有效，但若要最大化模型的准确性，需要寻找缺失数据中的模式。要做到这一点，可以观察特征的联合分布，例如，同一组的乘客是否倾向于来自同一家庭？显然存在许多组合，因此我们将总结我和其他人发现的有用趋势。因此，本文接下来会对这些联合分布进行研究。

联合分布分析

首先是对于 HomePlanet 的填补。

Group 和 HomePlanet 的联合分布如下表所示（仅展示前 5 组）：

HomePlanet	Earth	Europa	Mars
Group			
1	0	1	0
2	1	0	0
3	0	2	0
4	1	0	0
5	1	0	0

表 12 联合分布表结果示例

接着，对每一组的不同星球个数进行计数做统计，发现每一组都是来自于同一个星球，没有例外。所以，可以按照组别来补充其中来自星球的信息。

经过填补之后，HomePlanet 属性的缺失值从 288 降低至 157。

HomePlanet 和 CabinDeck 的联合分布如下热力图所示：

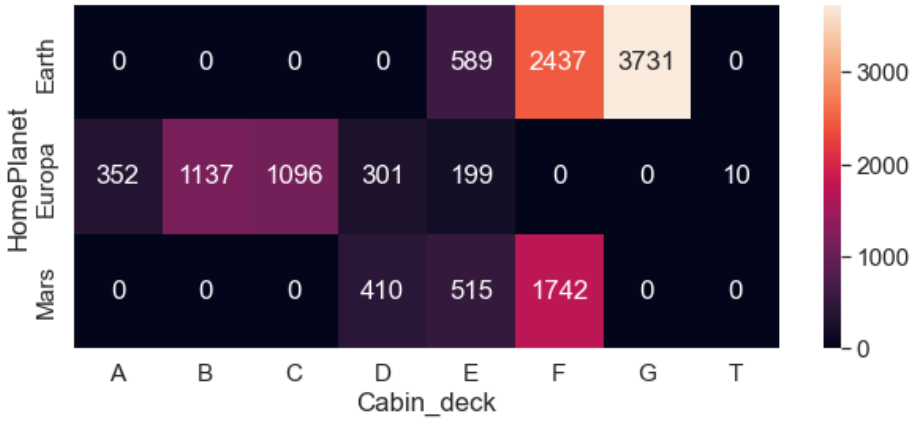

图 13 HomePlanet 和 CabinDeck 的联合分布热力图

可以得出以下结论：

a) 来自 Europa 的乘客在 A、B、C 或 T 层甲板上。

b) 来自 Earth 的乘客在 G 层甲板上。

c) 来自多个行星的乘客在 D、E 或 F 层甲板上。

利用这样的一个规律可以将 A,B,C,T,G 甲板上母星缺失的给填补上。

经过填补之后，HomePlanet 的缺失值从 157 降低至 94。

HomePlanet 和 Surname 也有与 Group 和 HomePlanet 类似的关系，即同一个姓氏的人都来自于同一个星球，因此我们可以借此规律进行填补。填补后，HomePlanet 的缺失值从 94 降低至 10。

对于剩下的 10 个缺失值，经画表发现这 10 个缺失值的目的地都是 TRAPPIST-1e，所以这里研究 HomePlanet 和 Destination 之间的关系，绘制热力图如下所示：

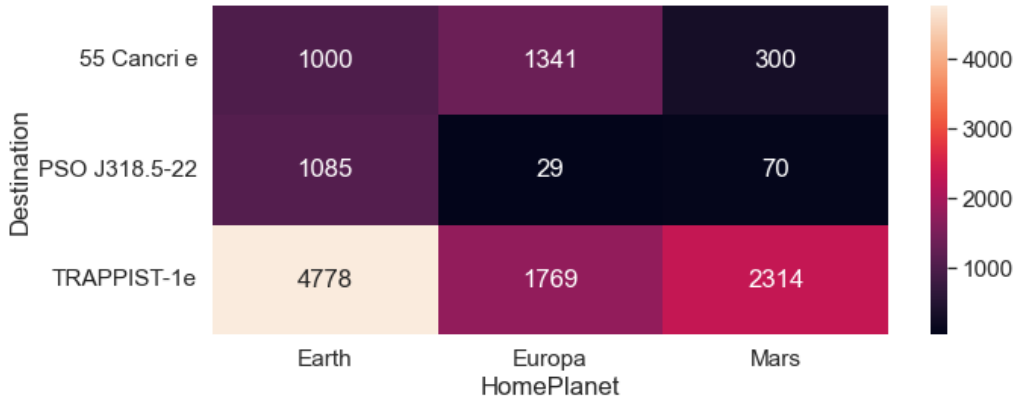

图 14 HomePlanet 和 Destination 的联合分布热力图

大多数前往 TRAPPIST-1e 的人来自地球，因此用 Earth 进行填补是合理的。没有人来自地球是在 D 层甲板上的，所以我们需要将它们过滤掉。因此最后将这 10 个缺失值填补完毕。至此，所有的 HomePlanet 的缺失值已经全部填补完毕。

接着是对 Destination 的填补。

对于上一个热力图分析可得，大部分的乘客的 Destination 均为 TRAPPIST-1e，所以这里采用众数填补法，将 Destination 的缺失值用 TRAPPIST-1e 填补。

再者是对 Surname 的填补。

探索 Surname 和 Group 的关系，绘制直方图如下所示：

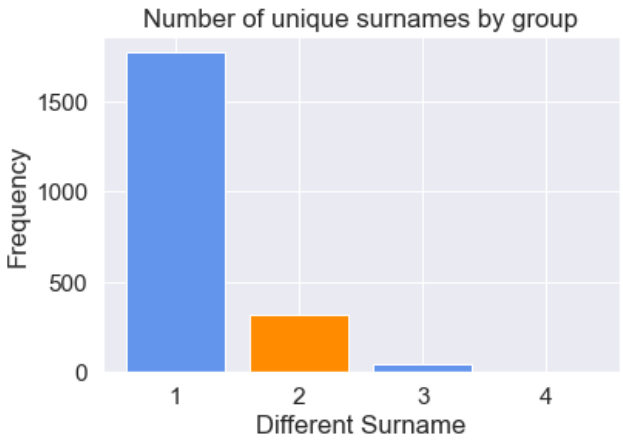

图 15 Surname 和 Group 关系直方图

大多数（83%）的群组只包含一个家庭。因此，可以根据该群组中的大多数姓氏来填充缺失的姓氏。填补后，Surname 的缺失值从 294 降低至 155。不必去除所有这些缺失值，因为最终会删除姓氏特征。然而，本文可以据此更新家庭大小特征。

接着是对 Cabin_side 的填补。探索 Cabin_side 和 Group 之间的关系，采用与探索 Surname 和 Group 的关系相同的方法，分析可得所有的相同组的人都在甲板相同的 side，因此可以利用这样一点来补充所在甲板的位置。填补之后，Cabin_side 的缺失值从 299 降低至 162。再次探索 Cabin_side 和 surname 之间的关系，画出，绘制出家庭成员在同一侧的甲板占比的直方图如下所示：

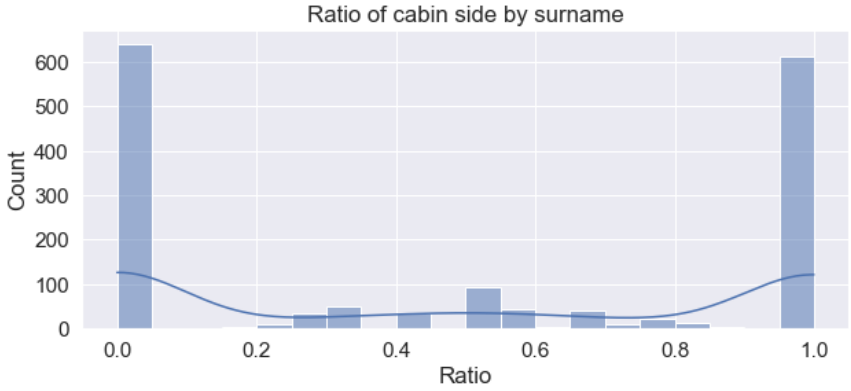

图 16 家庭成员在同一侧甲板占比直方图

分析可得，家庭成员在同一侧所占的百分比 76.7%，所以可以使用这一规律来填补缺失值。填补之后，Cabin_side 的缺失值从 162 降低至 66。由于 Cabin_side 是一个平衡的数值，剩余的数据无法草率的用其中一种去填补，所以这里全部采用异常值去填充。至此，所有的 Cabin_side 的缺失值填补完毕。

接着去填补 Cabin_Deck 的缺失值。首先有个前面分析出来的规律可以首先对缺失值进行初步填补，即来自相同家庭的成一半在同一个分组当中，利用这样的一个规律进行填补。填补之后，缺失值数量从 299 降低至 162。接着进行联合分布分析，得到如下表所示的结果：

HomePlanet	Destination	Solo	A	B	C	D	E	F	G	T
Earth	55 Cancri e	0	0	0	0	0	20	90	272	0
		1	0	0	0	0	47	289	269	0
	PSO J318.5-22	0	0	0	0	0	18	67	230	0
		1	0	0	0	0	25	262	466	0
	TRAPPIST-1e	0	0	0	0	0	133	438	1075	0
		1	0	0	0	0	358	1350	1509	0
Europa	55 Cancri e	0	96	377	313	59	35	0	0	2
		1	67	141	159	46	34	0	0	0
	PSO J318.5-22	0	2	5	11	0	0	0	0	0
		1	0	0	10	0	0	0	0	0
	TRAPPIST-1e	0	152	459	428	120	53	0	0	1
		1	44	179	201	84	82	0	0	8
Mars	55 Cancri e	0	0	0	0	32	15	104	0	0
		1	0	0	0	40	16	92	0	0
	PSO J318.5-22	0	0	0	0	8	9	14	0	0
		1	0	0	0	9	7	21	0	0
	TRAPPIST-1e	0	0	0	0	168	219	798	0	0
		1	0	0	0	164	263	743	0	0

表 13 HomePlanet、Destination、solo 联合分布表

分析上述数据可以得出如下结论：

a) Mars 的乘客最有可能在 F 层甲板上。

b) Europa 的乘客（大致上）如果是独自旅行，则最有可能在 C 层甲板上，否则可能在 B 层甲板上。

c) Earth 的乘客（大致上）最有可能在 G 层甲板上。

利用上述规律进行填补后，缺失值数量从 162 降低至了 0。至此 Cabin_Side 填补完毕。

接着对 Cabin_number 进行填补，分析 Cabin_number 和 Deck 之间的关系，绘制出以下的散点图：

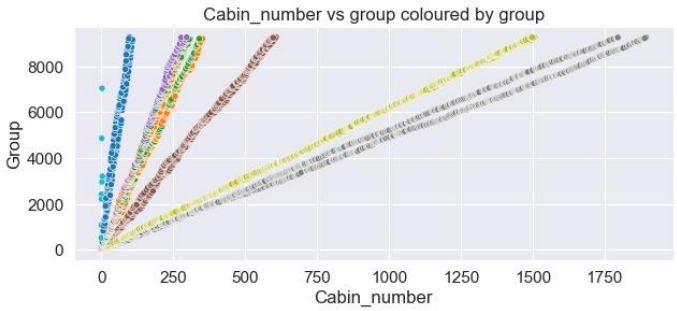

图 17 Cabin_number 和 Deck 关系散点图

舱位号和组号在每个甲板上共享线性关系。因此，这里可以通过在每个甲板上使用线性回归来推断缺失的舱位号，以获得一个近似的舱位号。填补后，Cabin_number 的缺失值从 299 降低至 0。

这里可以更新以下甲板分类特征工程的结果，即将填充好的 Cabin_number 分配到特征工程所做的以 300 为阶的分类结果中。

接着是对 VIP 进行填充。由于 VIP 是一个极其不平衡的量，大部分人都是非 VIP，所以采用众数填补法，即非 VIP 填充缺失值。

接着是对于年龄的填充。年龄在家乡星球、团队规模、花费和舱位甲板等许多特征上都存在变化，因此将根据这些子组的中位数来填补缺失值。在此，再将新填充的缺失值加入之前特征工程中的分组。

对于 CryoSleep 这个属性，如果一个乘客没有消费记录那他大概率会在冬眠，如果有消费记录，那在冬眠的可能性其实就不大。根据这个规律填充这个属性是合理的。

最后是对 Expenditure 进行填补。首先根据上述规律，即冬眠的人没有消费这个规律，将冬眠的样本用 0 进行填充，缺失值从 1410 降低至 866。对于剩余的 866 个缺失值，本文采用以下的策略：花费在许多特征上存在差异，但只会使用家乡星球、独自旅行

以及年龄组来填补缺失值，以防止过拟合。本文还将使用平均值而不是中位数，因为很大一部分乘客没有花费，中位数通常为 0。并且，年龄在 12 岁以下的人不会有任何花费。联合分布的表格如下所示：

HomePlanet	Solo	Age_0-12	Age_13-17	Age_18-25	Age_26-30	Age_31-50	Age_51+
Earth	0	0	724.9022222	789.7005545	841.0935961	736.6557734	733.6495726
	1	0	693.0148976	779.3959417	795.4206897	794.8186275	826.3669725
Europa	0	0	1153.160256	2652.013298	3534.668246	3975.774005	3483.639004
	1	0	0	2489.888889	3806	3949.939929	3952.085526
Mars	0	0	1176.839286	1161.808333	1247.098361	1143.671916	1345.419643
	1	0	1687.261538	1075.341146	1107.122677	1110.392045	1100.298387

表 14 HomePlanet、Solo、Age 联合分布表

通过分析这个表，我们可以发现一些有趣的趋势和规律。例如，对于年龄段为 18-25 岁的乘客，他们的平均旅游花费在所有星球中都最高，这可能是因为这个年龄段的乘客更愿意花费更多的钱来体验更多的旅游活动和设施。此外，我们还可以看到一些星球的旅游业务特征不同。例如，Europa 星球的平均旅游花费在所有星球中最高，这可能是因为 Europa 是一个相对较为偏远和神秘的星球，吸引了更多寻求冒险和探索的旅客。

采用平均数填补后，缺失值数量降为到 0。至此，所有缺失值填补完毕。

数据预处理

对于训练集和测试集，本文做了以下的预处理：

a) 删除'PassengerId', 'Group'两个属性，因为这两个数据在特征工程中，分别用于结对出行人的个数。

b) 删除 Cabin_number，因为此用于定性甲板的分组，这里已没有作用。

c) 删除 VIP，因为他对结果几乎没有影响。

对于消费类属性（即 RoomService, FoodCourt, ShoppingMall. Spa），他们的分布差距比较大，大部分样本集中在了少部分的区域中。本文采取对数变换的方法，减小分布的偏斜程度，尤其是这里还有特别大的异常值。

现在作出分布直方图如下：（这里仅展示 RoomService 和 FoodCount，其他规律基本相同）

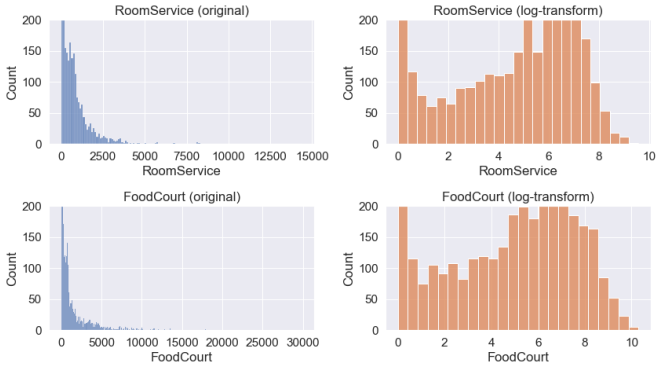

图 18 取对数后的数据分布直方图

可以看到经过对数变换之后分布明显变得比之前更加均匀。

接着对数据进行编码和合理的缩放处理。步骤如下：

a) 识别数值列和分类列；

b) 将数值数据缩放为均值 $=0$ 和方差 $=1$；

c) One-hot 编码分类数据；

d) 数值预处理，分类预处理，组合预处理（使用 ColumnTransformer）

经过这样的处理，可以将这些数据放入分类器进行训练。

主成分分析

使用 PCA 类对输入数据 X 进行拟合，获取每个主成分的解释方差比例。使用 matplotlib 库绘制折线图，横轴为主成分数量，纵轴为解释方差的累积比例。绘制一条红色的水平线，表示解释方差累积比例达到 100%的阈值。

方差累计曲线图如下（图 17）所示：

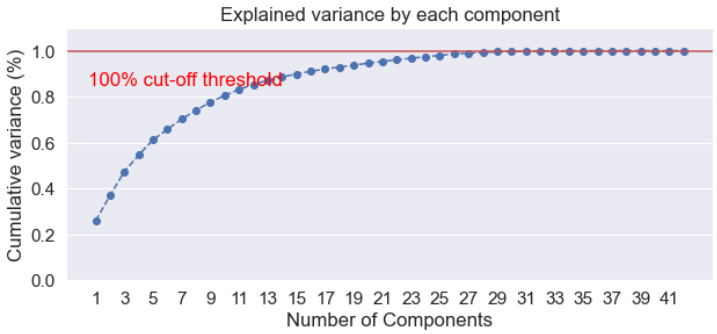

图 19 方差累计曲线图

对于方差累计曲线图分析可得，在 1-20 曲线处于上升阶段，表示前 20 个主成分包含了相对重要的信息。而 20 以后趋于平缓，说明更多的主成分对于方差的贡献较小，重要性相对更低。这对于以后分类器的参数设置有一定的参考价值。

模型建立和提交

首先使用 lazypredict 库中的模型尝试分类模型，得到的分类结果如下所示：

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score
LGBMClassifier	0.80	0.80	0.80	0.80
AdaBoostClassifier	0.79	0.79	0.79	0.79
RandomForestClassifier	0.79	0.79	0.79	0.79
XGBClassifier	0.79	0.79	0.79	0.79
NuSVC	0.78	0.78	0.78	0.78
SVC	0.78	0.78	0.78	0.78
RidgeClassifierCV	0.78	0.78	0.78	0.78
LinearDiscriminantAnalysis	0.77	0.77	0.77	0.77

表 15 各个模型性能差异表

这里仅展示准确度前 8 的模型。

通过这张表，我们可以看到不同算法的表现都比较接近，性能指标得分在 0.77 至 0.80 之间。其中，LGBMClassifier 算法表现最好，但其他算法的表现也很不错，例如 AdaBoostClassifier、RandomForestClassifier 和 XGBClassifier 等。

本文又尝试了 catClassifier 的模型，这里尝试 catClassifier 的模型得到的准确率也在 0.80。

从 lazypredict 的结果看来三种效果最好的模型是 cat，LGBM，XGB，AdaBoost 四种模型，我们将尝试用四种模型进行集成学习投票来决定。

本文分别采取了软投票和硬投票的方式，得到的结果如下表所示：

投票方式	Accuracy
Soft(软投票)	0.80506
Hard(硬投票)	0.79873

表 16 硬投票和软投票性能差异

可以发现，相对于单独的模型，采用软投票集成学习的方法准确率有所提高。

为了进一步尝试更多的投票方式，本文还尝试了调整各个分类器的权重进行投票，得到的结果如下表所示：

权重占比（lgbm, xgb, ada:cat）	Accuracy
1, 1, 1, 1	0.80103
2, 1, 1, 1	0.80104

表 17 不同权重投票结果差异

相对于单个模型略有提升，但是仍不如软投票的方式。

针对上述探索的规律，可以发现 catClassifier 的模型准确率相对较高，所以对于这两个模型进行单独调参尝试，使用 GridSearch 的函数进行尝试，其中参数的设置如下表所示：

参数名称	参数值
depth	4, 6, 8,10
learning_rate	0.03, 0.1, 0.15, 0.20, 0.26
l2_leaf_reg	1, 4, 9, 16
iterations	300

表 18 对 catClassifier 单独调参的参数表

得到的最优模型的 accuracy 的结果为 0.80966，是目前为止最高的。

接着，本文还尝试了使用 stack 堆叠的方法对模型进行进一步优化，本文设置的参数如下所示：

参数名称	参数值
final_estimator__depth	4, 6, 8
final_estimator__learning_rate	0.03, 0.1, 0.15
final_estimator__l2_leaf_reg	1,4,9
final_estimator__iterations	300

最后得到的精确度为 0.80851。

综上所述，本文尝试的方法有软投票，硬投票，更改权重的投票，对 catClassifier 单独调参优化，stack 堆叠法优化，制作如下的图进行可视化：

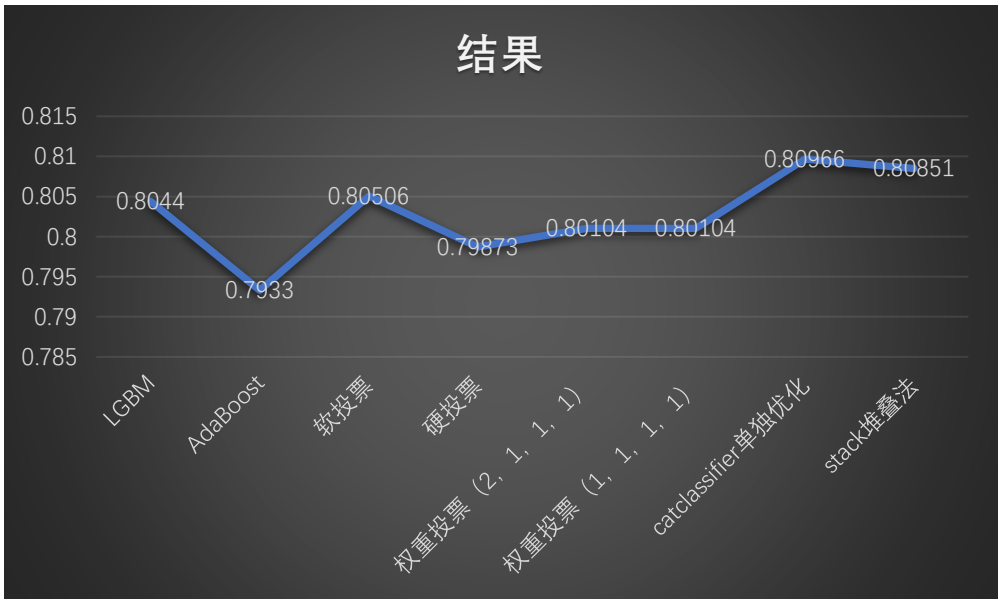

图 20 各优化方法效果对比

可以发现，最优的方法在 catclassifier 单独优化和 stack 之间，所以保存着两个模型，提交至 kaggle 官方。

提交至 kaggle 后，最后的模型准确率为 0.80827，排名 8%（234/2637）。

综上所述，模型在经过改进后，提升显著。

总结

本文针对太空船泰坦尼克号的乘客数据集，首先尝试**最基础**的方法，即对数据类型进行分析后，采用最基本的方式进行缺失值填补，比如**众数**和**均值**填充法。数据的预处理也采用机器学习中最基本的方法将非数值型转换为数值型。这种粗糙的处理导致很多隐藏在数据集中的特征被埋没了，所以最后的结果不尽如人意。本文还尝试对模型进行改进，首先对数据进行**探索性分析**，根据数据类型的不同，探索数据中的特征，发现了一些隐藏的特征，比如传送的**成功率和年龄分布有关，CryoSleep 和是否消费紧密相关，**甚至还发现了 **name 可以确定一个人的家族，ID 可以确定成队出行的人数**。在做**特征工程**的时候，就可以灵活的采用探索性数据分析中的结论，比如**将年龄分段，将所有的消费属性归为一类消费**，抑或是**根据 ID 来对乘客组别大小进行特征提取**等等。接着根据特征工程做缺失值处理，此时很多的处理便不像一开始的均值和众数如此粗糙，精确度进一步提升。最后在模型建立的时候，还用了**投票法**和精细**调参**对模型进一步优化。对于本次任务总结可得，对于这种不是很大的数据集，采用什么分类器或者对分类器进行

调参优化仅仅是锦上添花，因为根据 kaggle 的排行榜可以看出仅仅对于分类器进行优化的队伍仅仅准确率只能达到 78%左右；本次任务的关键在于对数据中的细微特征的提取，从 0%到 78%其实很简单，但从 78%到 80%是那艰难的“最后一公里”，考验的也正是对于数据探索和特征提取的细致性以及方法运用的灵活性，也正是本文的最大亮点。也正因此，本次任务从一开始的 1921/2637 提升至 234/2637，达到了 8%排名的好成绩。

参考文献

[1] 李航. (2019). 统计学习方法（第 2 版）. 清华大学出版社.

[2] 周志华. (2016). 机器学习. 清华大学出版社.

[3] Shaposhnikov, D., & CatBoost Development Team. (2018). CatBoost: unbiased boosting with categorical features support. arXiv preprint arXiv:1810.11363.

[4] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems (pp. 3146-3154).

[5] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).