泰坦尼克号：从灾难中学习机器

孙浩然 17182627 王润欣 17182625 李子易 17182624

（北京航空航天大学北京学院 北京 100191）

Titanic : Machine Learning from Disaster

Sun Haoran, Wang Runxin , and Li Ziyi

[1]（*College of Beijing, Beihang University, Beijing* 100191）

Abstract Titanic’s silence is one of the most regrettable shipwrecks in history. On April 15, 1912, the Titanic, widely regarded as "never sunk", sank after colliding with an iceberg during the maiden voyage. Unfortunately, there were not enough lifeboats on board, resulting in the death of 1502 of the 2224 passengers and crew. Although survival requires some luck, some people are more likely to survive than others. We built SVM classifiers, random forest models, and Catboost classifiers through passenger data such as "name, age, gender, and socioeconomic class." We tried to predict who is more likely to survive through n cross-validations.

Key words Random Forest model; n fold cross validation

摘要 泰坦尼克号的沉默是历史上最让人惋惜的沉船事件之一。1912 年 4 月 15 日，被广泛认为是“永不沉没”的泰坦尼克号在处女航中与冰山相撞后沉没。不幸的是，船上没有足够的救生艇，导致 2224 名乘客和船员中有 1502 人死亡。虽然生存需要一些运气因素，有些人群比其他人更有可能生存下来。我们通过“姓名，年龄，性别，社会经济阶层”等乘客数据建立了 SVM 分类器，随机森林模型，Catboost 分类器，通过 n 次交叉验证来尝试预测到底有哪些人存活下来的可能性更高。

关键词 随机森林模型；n 折交叉验证

1. 任务背景

1. 赛题任务

泰坦尼克号的沉没是历史上最为声名狼藉的沉船事件之一。1912 年 4 月 15 日，被广泛认为是“永不沉没的”泰坦尼克号在处女航中与冰山相撞后沉没。不幸的是，船上没有足够的救生艇，导致 2224 名乘客和船员中有 1502 人死亡。虽然生存需要一些运气因素，但似乎有些人群比其他人更有可能生存下来。在这个挑战中，我们要求您建立一个预测模型来回答这个问题:“什么类型的人更有可能生存？” 使用乘客数据(如姓名、年龄、性别、社会经济阶层等)。

2. 赛题数据

在此任务中数据被分为两组:

（1）训练数据将提供：

Survival: 人员是否存活（0,1）
Pclass: 船票等级（1,2,3）
Sex: 性别（male,female）
Age: 年龄
Sibsp: 兄弟姐妹/配偶数
Parch: 父母孩子数
Ticket: 门票号码
Fare: 乘客收费
Cabin: 仓位号码
Embarked: 启程港(C,Q,S)

（2）测试数据将提供：

Pclass: 船票等级（1,2,3）
Sex: 性别（male,female）
Age: 年龄
Sibsp: 兄弟姐妹/配偶数
Parch: 父母孩子数
Ticket: 门票号码

Fare: 乘客收费
Cabin: 仓位号码
Embarked: 启程港(C,Q,S)

2. 数据分析与处理

1. 数据预处理

我们先将训练数据集与测试数据集通过 python 的 pandas 进行了数据的导入并且将训练集与测试集进行了合并。之后对各个数据字段进行了缺失值的统计。结果如下：

```
[1309 rows x 12 columns]
Survived     418
Age          263
Fare           1
Cabin       1014
Embarked       2
dtype: int64
```

图 1

我们发现对于船舱位置（cabin）和年龄（age）特性的缺失值比较多。为了进一步研究各个特征对于乘客是否生存的影响，此处先进行了简单的平均值填充缺失值的方法将各个字段进行了缺失值处理。当然，这并不是我们最终对缺失值的处理，此处进行简单的缺失值填充是为了方便我们进一步对数据进行分析。处理后缺失情况如下：

```
Survived    418
dtype: int64
```

图 2

2. 数据分析

在对数据进行了简单的预处理以后，现在的数据已经比较完整了。这方便了我们对数据做一些简单的可视化显示。所以，我们将各个特征值与生存的关系进行了简单的数据可视化。如下：

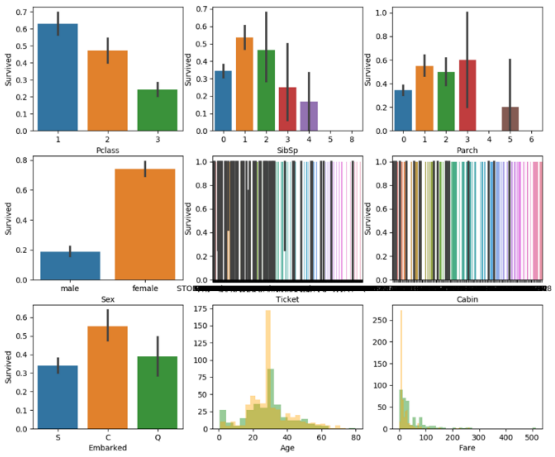

图 3

先总体来看的话，我们明显发现船票（ticket）和船舱位置（cabin）两个字段的特征并不明显，导致其与乘客是否存活的关联性并不能够明显看出。之后，我们再来仔细地分析每一个特征对乘客是否生存的影响程度。

对于船票等级（pclass），我们对该特征进行了进一步的可视化。如图 4 所示：

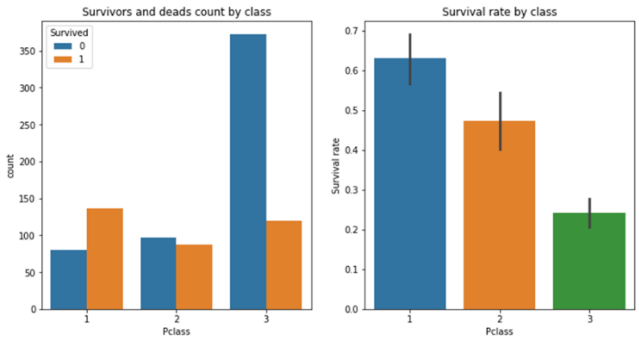

图 4

我们先分别对 pclass 为 1,2,3 的等级的中乘客的存活人数做了统计并可视化显示，另外还对该特征与存活率之间的关系做了具体显示。从图中我们不难发现对于船舱等级为 1 的乘客，我们发现其中存活的乘客甚至比死亡的乘客要多。在通过观察船舱等级为 1 的乘客的存活率，发现其存活率高达 60%以上。所以我们猜测船舱等级数越小乘客的船舱应该是越好，越高级，导致其存活率会比较高。为了验证我们的猜想，我们继续分析了船舱等级为 2 和 3 的乘客的存活情况，先做内部比较发现，等级为 2 的乘客的存活于死亡大致持平，死亡略高于存活人数，存活率大概也在 48%左右。而再观察等级为 3 的乘客，发现死亡人数是存活人数的三倍多，而存活率也是低到了 28%左

右。所以，我们不难总结出，船舱等级越高的乘客（数越小），越容易存活，而船舱等级比较低的乘客不易存活。这可以证明 pclass 特征对乘客是否生存的影响是巨大的，所以这是我们不可忽视的特征。

接着，我们又对性别（sex）特征进行了分析。同 pclass 特征的处理方式，这里我们也是对数据进行了具体的可视化显示（如图 5）：

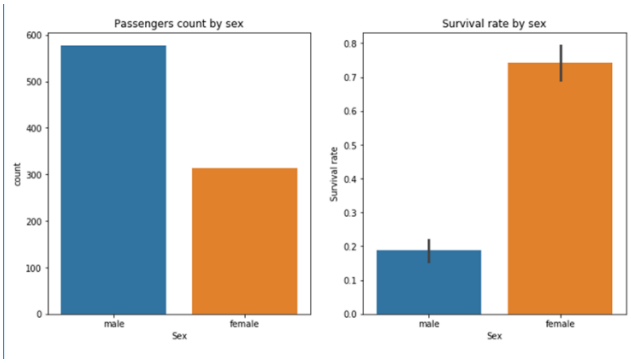

图 5

为了避免男性和女性数据量的差异，所以我们对数据中的男女数量做了一个简单的统计。我们发现男性人数大概是女性人数的两倍，男性人数在 600 左右，而女性人数在 300 左右。再观察男性乘客与女性乘客的存活率，我们发现女性乘客虽然人数少，但是女性的存活率是男性的三倍多，所以我们可以确定性别对乘客存活情况的影响很大。这也是我们需要重视的数据，会对我们的预测结果产生很大的影响。

之后，我们又对船票价格（fare）进行了具体的分析。并做了进一步的可视化分析。如图 6 所示：

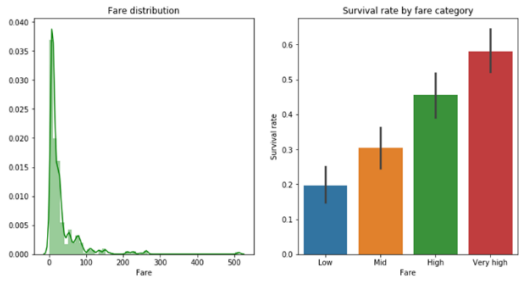

图 6

我们发现乘客的船票的价格分布大概在 30 左右，说明大部分人的船票价格偏低，而只有少一部分人的船票价格比较高。为了方便做数据的分析，我们又将船票价格从最低到最高进行了 4 等分，做了简单的分类，分成 low，mid，high，very high 四类来代表船票价格等级。我们可以从图中发现，船票高的乘客的存活率会高，而且船票价格由低到高，乘客的存活率也是逐渐增长的。这证明了船票价格对乘客是否存活的影响程度很大，所以这一特征会对我们有所帮助。

我们对年龄（age）特征也进行了类似的相关分析与数据可视化。但考虑到年龄的数据缺失值比较多，而我们又是采取的平均值填充的方法去做的预处理，所以参考的价值相对前几个特性应该不是那么的大。具体数据可视化如图 7 所示：

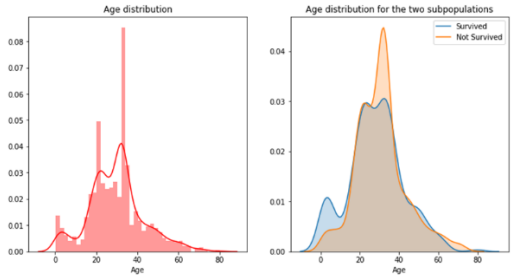

图 7

左边的图描述了乘客年龄的分布，右边的图描述了年龄与存活率的关系。我们发现乘客中的中年人比较多，而存活情况是年幼的孩子存活率比较高，超过了死亡率，而 30 岁左右的中年人存活率很低，60 岁以上的老人存活率也不是很高。该特征值也会在之后的预测中产生一定的影响。

然后，我们又对港口号（Embarked）进行了数据的分析和可视化的展示，如图 8：

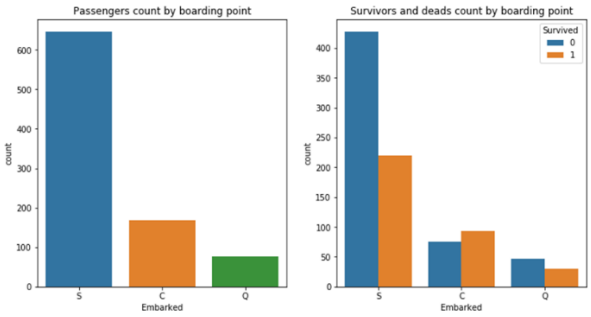

图 8

我们发现 s 港口的人数比较多，C 和 Q 港口的人数比较少。再通过观察各个港口的乘客的存活情况，我们发现，各个港口的存活情况差距也巨大。从 s 港等船的人最多，但是存活率却是最低的，而从 Q 港登船的人虽然最少，但是存活率也不是很高，明显低于 50%。反而是登船人数居中的 C 港的存活情况比较乐观，存活人数大于死亡人数，存活率在 50%以上。这使得我们对该特征与乘客存活的联系产生了很大的疑惑。因为只从港口号和乘客生存率来看，我们并不能得出什么结论。考虑到实际情况中，登船港口与地理位置有关系，而地理位置往往决定着一个人的职位与阶级，所以我们猜测港口号可能受到了其他特征的影响，然后我们又猜测应该是船舱等级（pclass）特征对港口号产生了一定程度的影响。于是，我们又分

析了从各个港口登船的人的船舱等级分布。如图 9 所示：

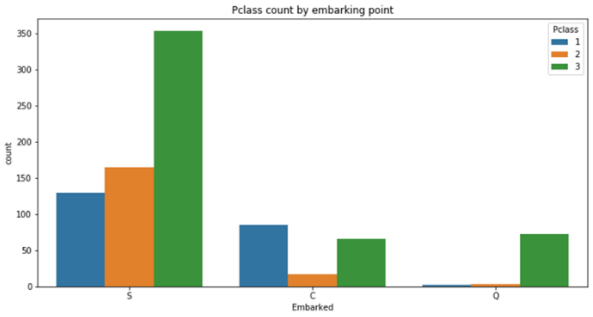

图 9

果然像我们所猜测的那样，从图中可以看出，从 s 港，Q 港登船的人中，船舱等级为 3 的人占比高，而 C 港登船的人中，船舱等级为 1 的人占比较高。所以此特征也可以作为辅助我们进行乘客是否存活的预测。

最后，我们也发现了在名字（name）字段中隐藏着一些重要的信息，这些信息往往代表着乘客的一些相关特征。这会再之后的数据处理中做详细的描述。

3. 数据修改

现在，我们准备挖掘出名字（name）中的有用信息。首先，我们将 name 字段用逗号和空格分隔开，选择第一个项信息：

```
train['title']=train.Name.apply(lambda x: x.split('.')[0].split(',')[1].strip())
test['title']=test.Name.apply(lambda x: x.split('.')[0].split(',')[1].strip())
```

并显示出提取出的 title 数量与年龄关系(图 10、11)：

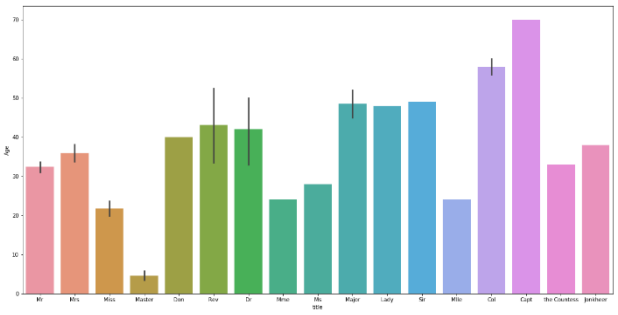

图 10

```
Mr              517
Miss            182
Mrs             125
Master           40
Dr                7
Rev               6
Mlle              2
Col               2
Major             2
Capt              1
Don               1
the Countess      1
Jonkheer          1
Ms                1
Sir               1
Lady              1
Mme               1
Name: title, dtype: int64
```

图 11

我们发现这些名字中所带的描述特征的信息与年龄有着一定的联系(图 12)，并且可以提取与职业、阶级有关的信息作为幸存预测的特征。

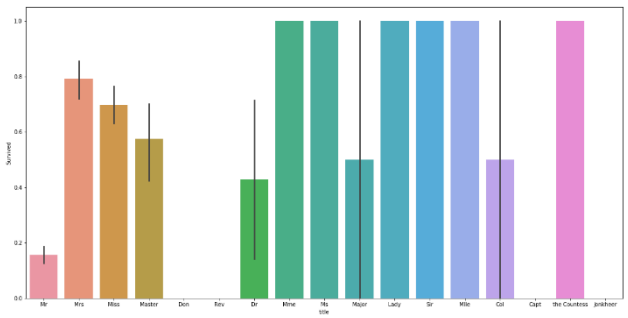

图 12

从图中我们可以看出名字中带有“Mr”的乘客幸存率较低，而“Mrs”、“Miss”等存活率较高，还有一些少数职业如“Capt”的存活率很高。我们将这 18 个特征化为 4 个更为有特点的新特征(图 13)。

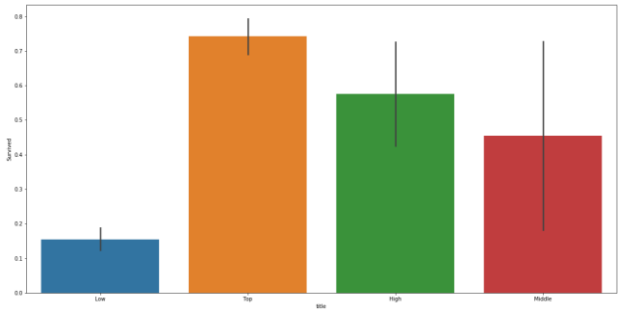

图 13

对于 Cabin(船舱号)这个特征，缺失了太多的数据，但是对有限的数据进行分析，我们发现乘客的幸存与否与此特征有一定联系，并且在现实中船舱号也是判断生存率的一项重要特征，所以我们将缺失数据填为“U”，其他已知数据改为其首字母(图 14)。

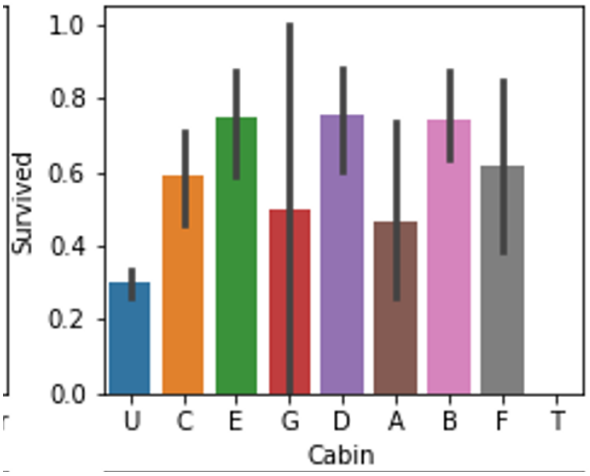

图 14

3. 模型构建

1. 随机森林

(1)原理介绍

要对一个输入的样本进行分类，将这个样本交给每棵树进行分类，根据所有树分类的结果选择分成哪一类。而每棵树的产生规则是在大小为 N 的训练集中有放回地抽取 N 个训练样本，作为该树的训练集，有放回和随机抽样保证了差异性与准确性。在得到了训练集后选取特征数开方个特征，在每次树进行分裂时从这些特征中选择最优的，可用算法有 ID3、C4.5、C5.0 等，每棵树都尽可能地生长不会进行剪枝。

随机选取训练集和随机选取特征成为了随机森林分类性能良好的主要条件，使得随机森林不易陷入过拟合，并且有很好的抗噪能力。

(2)代码实现

调用 sklearn 包中的随机森林分类器实现分类，具体代码如下：

```
RF=RandomForestClassifier(random_state=None)
PRF=dict(max_depth = [9],
         min_samples_split = [11],
         min_samples_leaf = [6],
         n_estimators = [40])
GSRF=GridSearchCV(estimator=RF, param_grid=PRF, scoring='accuracy',cv=10)
scores_rf=cross_val_score(GSRF,xtrain,ytrain,scoring='accuracy',cv=10)
print("rf : ",np.mean(scores_rf))
```

其中参数部分(PRF)是已经通过 GridSearchCV 选出的较好参数，并没有将开始的备选参数展示出来。模型测试部分采用 10 折交叉验证取平均的方式，对训练集进行评估，在采用下图中特征为分类标准时结果如下：

```
['Survived' 'Pclass' 'Sex' 'Age' 'Fare' 'FamilySize' 'Embarked_C'
 'Embarked_Q' 'Embarked_S' 'title_Master' 'title_Miss' 'title_Mr'
 'title_Mrs' 'title_Officer' 'title_Royalty']
rf :  0.8406327318125072
```

可见对于本地训练集可以达到 84%的正确率，若采用以 F1 分数作为评分标准则可达到 0.77 分。

```
scores_rf=cross_val_score(RF,xtrain,ytrain,scoring='f1',cv=10)
```

```
['Survived' 'Pclass' 'Sex' 'Age' 'Fare' 'FamilySize' 'Embarked_C'
 'Embarked_Q' 'Embarked_S' 'title_Master' 'title_Miss' 'title_Mr'
 'title_Mrs' 'title_Officer' 'title_Royalty']
rf :  0.7714477801886754
```

在最后的提交中，我们发现对于测试集的得分仅为 0.79904 分，通过随机森林模型和对数据的处理，我们成功预测了 79.9%的乘客的生存情况。

2. SVM

（1）简介

支持向量机（support vector machines, SVM）是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机；SVM 还包括核技巧，这使它成为实质上的非线性分类器。SVM 的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。SVM 的学习算法就是求解凸二次规划的最优化算法。一般 svm 有三种：

硬间隔支持向量机（线性可分支持向量机）：当训练数据线性可分时，可通过硬间隔最大化学得一个线性可分支持向量机。

软间隔支持向量机：当训练数据近似线性可分时，可通过软间隔最大化学得一个线性支持向量机。

非线性支持向量机：当训练数据线性不可分时，可通过核方法以及软间隔最大化学得一个非线性支持向量机。在本文中，对于乘客是否生存的预测，是一个标准的二分类问题，所以考虑选择 Svm 分类器实现。

（2）原理介绍

超平面：定义为w^Tx+b=0，是能够将高维空间一分为二的平面。

要对一个输入的样本进行二分类，就相当于依据样本建立一个高维空间，然后利用 Karush-Kuhn-Tucker (KKT)条件和拉格朗日公式来推出边际最大化的划分超平面。对于线性不可分的数据可以利用内积核函数的方法来实现 svm 分类。同时， SVM 的最终决策函数只由少数的支持向量所确定,计算的复杂性取决于支持向量的数目,而不是样本空间的维数,这在某种意义上避免了“维数灾难”。

（代码实现）

调用 sklearn 包中的 svm 分类器实现分类，具体代码如下：

```
svc=make_pipeline(StandardScaler(),SVC(random_state=1))
r=[0.0001,0.001,0.1,1,10,50,100]
PSVM=[{'svc__C':r, 'svc__kernel':['linear']},
      {'svc__C':r, 'svc__gamma':r, 'svc__kernel':['rbf']}]
GSSVM=GridSearchCV(estimator=svc, param_grid=PSVM, scoring='accuracy', cv=2)
scores_svm=cross_val_score(GSSVM, xtrain.astype(float), ytrain,scoring='accuracy', cv=5)
print(np.mean(scores_svm))
```

其中对于参数的调节问题是通过 GridSearchCV 选出来的比较好的参数。模型的猜测部分同样采取了交叉验证的方式对训练集进行了评估。本地测试结果如下：

```
svm :  0.8249622045920489
```

可见对于本地训练集可以达到 82%的正确率，在最后的提交中，我们对于测试集的得分为 0.78468 分，通过 svm 模型和对数据的处理，我们成功预测了 78.4%的乘客的生存情况。

3. CatBoost

1）原理介绍

我们可以使用 CatBoost，而不需要任何显式的预

处理来将类别转换为数字。CatBoost 使用在各种统计上的分类特征和数值特征的组合将分类值转换成数字。

它减少了对广泛的超参数调优的需求，并降低了过度拟合的几率，这也导致了模型变得更加具有通用性。但它还包含一些参数，比如树的数量、学习速率、正则化、树的深度等等 CatBoost 算法的设计初衷是为了更好的处理 GBDT 特征中的 categorical features。在处理 GBDT 特征中的 categorical features 的时候，最简单的方法是用 categorical feature 对应的标签的平均值来替换。在决策树中，标签平均值将作为节点分裂的标准。这种方法被称为 Greedy Target-based Statistics，简称 Greedy TS，这种方法有一个显而易见的缺陷，就是通常特征比标签包含更多的信息，如果强行用标签的平均值来表示特征的话，当训练数据集和测试数据集数据结构和分布不一样的时候会出条件偏移问题。一个标准的改进 Greedy TS 的方式是添加先验分布项，这样可以减少噪声和低频率类别型数据对于数据分布的影响，添加先验项是一个普遍做法，针对类别数较少的特征，它可以减少噪声数据。对于回归问题，一般情况下，先验项可取数据集 label 的均值。对于二分类，先验项是正例的先验概率。利用多个数据集排列也是有效的，但是，如果直接计算可能导致过拟合。CatBoost 利用了一个比较新颖的计算叶子节点值的方法，这种方式(oblivious trees，对称树)可以避免多个数据集排列中直接计算会出现过拟合的问题

4. GridSearchCV 调参器

GridSearchCV 根据其名字可以拆分为两部分，GridSearch 和 CV，即网格搜索和交叉验证。这两个名字都非常好理解。网格搜索，搜索的是参数，即在指定的参数范围内，按步长依次调整参数，利用调整的参数训练学习器，从所有的参数中找到在验证集上精度最高的参数，这其实是一个训练和比较的过程。而 CV 则是交叉验证，交叉验证原理是将训练数据随机等分为 n 份，选取一份作为本次的测试集，剩下的数据作为本次的训练集预测测试集的结果，重复选出测试集和训练集，直到全部数据被成功预测一遍，得到了对训练集的预测结果。并重新将数据进行等分，重复上边的过程，再得到一份训练集的预测结果，重新等分操作进行 n 次，最后训练集的预测结果为这 n 次预测的平均值，与训练集真实结果进行对比，得到训练集的预测效果。

我们采用 GridSearchCV 的目的是减少调参所花的时间，将更多的精力放在对数据的分析、处理与选择上，以达到更好的预测效果。

4. 总结与分工

在本次任务中，随机森林的表现最好，成功正确预测了测试集 79.9%的乘客的生成情况。但是我们只是单独地使用了三个模型，并没有将它们整合到一起，若是使用软投票或是硬投票的方式综合考虑三个模型预测的结果，我们相信预测效果会更好。通过本次课程，我们学到了很多有关人工智能的知识，而这次的任务更是让我们有了实践的机会，参加 kaggle 这种外国知名的比赛，开阔了我们的视野，一次次地提交与修改代码令我们更加深刻的意识到自己的不足，这也成为了我们不断前进的动力。

在分工方面，对于报告的撰写由我们三位同学共同完成，基本占比一样。在代码部分我们一个人负责一个模型，孙浩然负责随机森林，王润欣负责 Svm，李子易负责 Catboost，数据的读入和基本处理由孙浩然完成，在对数据进行进一步处理与分析时由三个人共同完成。