杭州电子科技大学

《数据挖掘课程设计》

课 程 设 计 报 告

专　　业	信息与计算科学
班　　级	19071232
学生姓名	张艺泷
学　　号	19071232
指导教师	邵新平
实验地点	6 教 402
完成日期	2022 年 6月 29日

目录

1. 背景

1.1 数据挖掘背景

1912 年 4 月 15 日，在她的处女航中，被广泛认为是“永不沉没”的皇家邮轮泰坦尼克号在与冰山相撞后虽然生存中有一些运气因素，但似乎有些群体比其他群体更有可能生存下来。在本次挑战中，要求建立一个预测模型，以回答以下问题：“什么样的人更有可能存活?”使用乘客数据（即姓名、年龄、性别、社会经济阶层等）。

1.2 数据集介绍

数据分为两组：训练集（train.csv）测试集（test.csv）训练集应用于构建机器学习模型。对于训练集，我们为每位乘客提供结果。您的模型将基于乘客性别和阶级等“特征”。您还可以使用特征工程来创建新特征。测试集应用于查看模型在未看到的数据上的性能。

	PassengerId	Survived	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked_C	Embarked_Q	...	Cabin_C	Cabin_D	Cabin_E	Cabin_F	Cabin_G	Cabin_T	Cabin_U	Family_Single	Family_Small	I
0	1	0.0	1	22.000000	1	0	A/5 21171	7.2500	0	0	...	0	0	0	0	0	0	1	0	1	
1	2	1.0	0	38.000000	1	0	PC 17599	71.2833	1	0	...	1	0	0	0	0	0	0	0	1	
2	3	1.0	0	26.000000	0	0	STON/O2. 3101282	7.9250	0	0	...	0	0	0	0	0	0	1	1	0	
3	4	1.0	0	35.000000	1	0	113803	53.1000	0	0	...	1	0	0	0	0	0	0	0	1	
4	5	0.0	1	35.000000	0	0	373450	8.0500	0	0	...	0	0	0	0	0	0	1	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1304	1305	NaN	1	29.881138	0	0	A.5. 3236	8.0500	0	0	...	0	0	0	0	0	0	1	1	0	
1305	1306	NaN	0	39.000000	0	0	PC 17758	108.9000	1	0	...	1	0	0	0	0	0	0	1	0	
1306	1307	NaN	1	38.500000	0	0	SOTON/O.Q. 3101262	7.2500	0	0	...	0	0	0	0	0	0	1	1	0	
1307	1308	NaN	1	29.881138	0	0	359309	8.0500	0	0	...	0	0	0	0	0	0	1	1	0	
1308	1309	NaN	1	29.881138	1	1	2668	22.3583	1	0	...	0	0	0	0	0	0	1	0	1	

1309 rows × 32 columns

图 1　数据集介绍

2. 数据预处理

2.1 缺失值处理

缺失值处理使用 pd.fullna 函数, 如果是数值类型，用平均值取代；如果是分类数据，用最常见的类别取代；使用模型预测缺失值，例如：K-NN。

对于年龄和船票价格，采用的是平均数来填充缺失值。对于登船港口，分别计算出各个类别的数量，采用最常见的类别进行填充。

对于船舱号，由于缺失的数据太多，将缺失的数据用’U’代替，表示未知。

2.2 数据分类

数据分类的过程比较麻烦，对于有直接类别的数据还有字符串类型的数据进行了不同方式的处理。

乘客性别（Sex）：男性 male，女性 female，令男性为 1，女性为 0

对于登船港口，客舱等级，客舱号，使用 pd.get_dummies 进行独热编码，列名前缀是 Embarked。

	Master	Miss	Mr	Mrs	Officer	Royalty	Pclass_1	Pclass_2	Pclass_3	FamilySize	...	Cabin_C	Cabin_D	Cabin_E	Cabin_F	Cabin_G	Cabin_T	Cabin_U	Embarked_C	Embarked_Q	Embarked_S
126	0	0	1	0	0	0	0	0	1	1	...	0	0	0	0	0	0	1	0	1	0
354	0	0	1	0	0	0	0	0	1	1	...	0	0	0	0	0	0	1	1	0	0
590	0	0	1	0	0	0	0	0	1	1	...	0	0	0	0	0	0	1	0	0	1
509	0	0	1	0	0	0	0	0	1	1	...	0	0	0	0	0	0	1	0	0	1
769	0	0	1	0	0	0	0	0	1	1	...	0	0	0	0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
316	0	0	0	1	0	0	0	1	0	2	...	0	0	0	0	0	0	1	0	0	1
792	0	1	0	0	0	0	0	0	1	11	...	0	0	0	0	0	0	1	0	0	1
247	0	0	0	1	0	0	0	1	0	3	...	0	0	0	0	0	0	1	0	0	1
757	0	0	1	0	0	0	0	1	0	1	...	0	0	0	0	0	0	1	0	0	1
724	0	0	1	0	0	0	1	0	0	2	...	0	0	1	0	0	0	0	0	0	1

图 2　数据预处理

3. 模型训练与评价

对于模型训练，使用 sklearn 库中的模型对数据进行训练。采用逻辑回归，随机森林，支持向量机，梯度提升决策，KNN，朴素贝叶斯分类多种算法进行训练并对其训练效果进行评价。

3.1 逻辑回归

3.1.1 模型原理

二分类问题: 对样本点 x，其标签 $y \in \{0,1\}$，想要知道的是概率

$$\Pr(y=1 \mid x) \tag{1}$$

对于二分类问题，显然可以建模假设 $y \mid x$ 服从参数为 p 的 Bernoulli 分布，因此只需要估计 $p=E(y \mid x)$

Bernoulli 分布是二项分布 n=1 的特殊情形，显然也属于指数分布族。使用 GLM 建模：选择 Bernoulli 分布 + 正规连接，得到

$$p=\frac{1}{1+e^{-\beta^T x}}=s\left(\beta^T x\right) \tag{2}$$

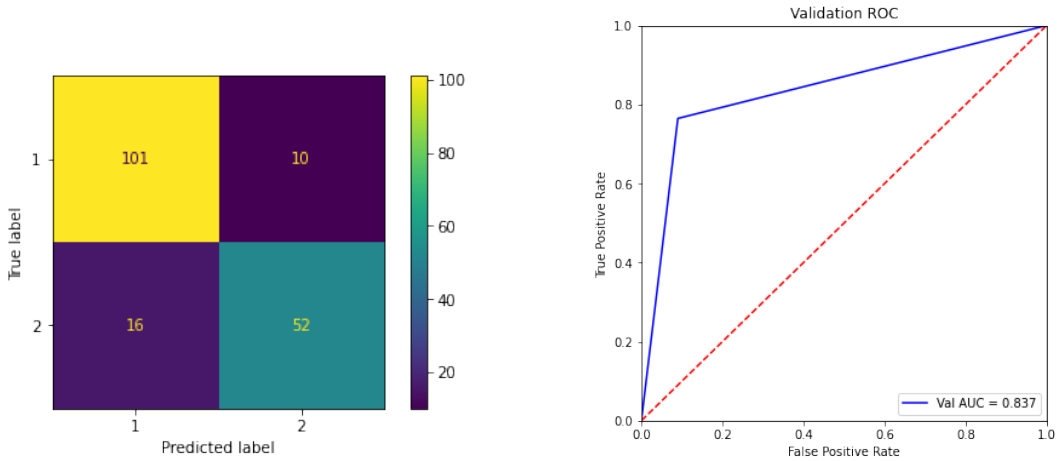
其中 $s(\cdot)$ 称为 Sigmoid 函数

$$s(z)=\frac{1}{1+e^{-z}} \tag{3}$$

3.1.2 模型结果

经过运行，当训练集比测试集为 8:2 时，可以得到逻辑回归 AUC=0.857。

```
#逻辑回归
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit( train_X , train_y )
```



(a) (b)

图 3　结果可视化

3.2 随机森林

3.2.1 模型原理

生成单棵决策树：（1）训练总样本的个数为 N，则单棵决策树从 N 个训练集中有放回的随机抽取 n 个作为此单颗树的训练样本。

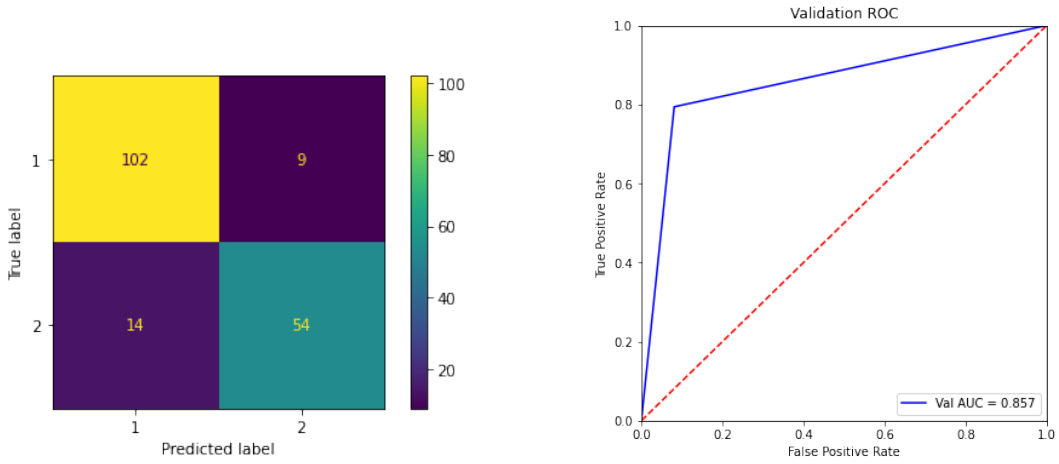
（2）令训练样例的输入特征的个数为 M，m 远远小于 M，则我们在每颗决策树的每个节点上进行分裂时，从 M 个输入特征里随机选择 m 个输入特征，然后从这 m 个输入特征里选择一个最好的进行分裂。m 在构建决策树的过程中不会改变。这里注意，要为每个节点随机选出 m 个特征，然后选择最好的那个特征来分裂。

（3）每棵树都一直这样分裂下去，直到该节点的所有训练样例都属于同一类。不需要剪枝。由于之前的两个随机采样的过程保证了随机性，所以就算不剪枝，也不会出现 over-fitting。

3.2.2 模型结果

经过运行，当训练集比测试集为 8:2 时，可以得到逻辑回归 AUC=0.837。

```
#随机森林Random Forests Model
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100)
model.fit( train_X , train_y )
```



(a) (b)

图 4　结果可视化

3.3 支持向量机

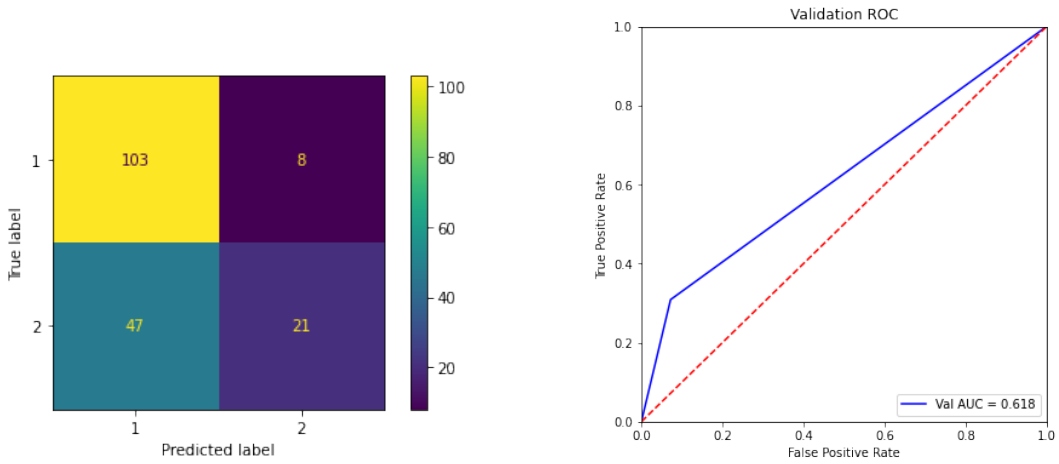
3.3.1 模型原理

SVM 是一种二类分类模型。它的基本模型是在特征空间中寻找间隔最大化的分离超平面的线性分类器。· 当训练样本线性可分时，通过硬间隔最大化，学习一个线性分类器，即线性可分支持向量机; · 当训练数据近似线性可分时，引入松弛变量，通过软间隔最大化，学习一个线性分类器，即线性支持向量机; · 当训练数据线性不可分时，通过使用核技巧及软间隔最大化，学习非线性支持向量机。硬间隔最大化 (几何间隔)、学习的对偶问题、软间隔最大化 (引入松弛变量)、非线性支持向量机（核技巧)。

3.3.2 模型结果

经过运行，当训练集比测试集为 8:2 时，可以得到逻辑回归 AUC=0.618。

```
#支持向量机Support Vector Machines
from sklearn.svm import SVC
model = SVC()
model.fit( train_X , train_y )
```



(a) (b)

图 5　结果可视化

3.4 梯度提升决策

3.4.1 模型原理

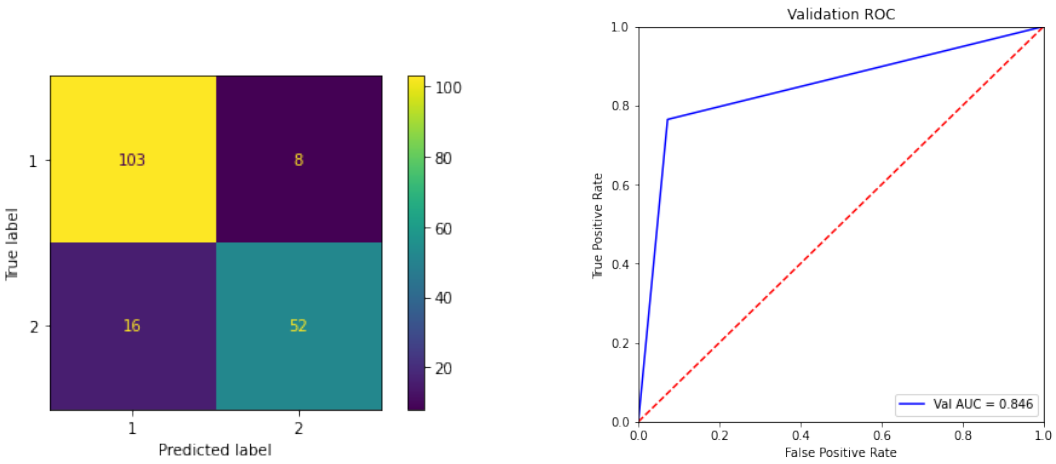
GBDT 是通过采用加法模型（即基函数的线性组合），以及不断减小训练过程产生的残差来达到将数据分类或者回归的算法，Friedman 提出了利用最速下降的近似方法，利用利用损失函数的负梯度在当前模型的值，作为回归问题中提升树算法的残差的近似值，拟合一个回归树。

$$-\left[\frac{\partial L\left(y_i, F\left(\mathbf{x_i}\right)\right)}{\partial F\left(\mathbf{x_i}\right)}\right]_{F(\mathbf{x})=F_{t-1}(\mathbf{x})} \tag{4}$$

3.4.2 模型结果

经过运行，当训练集比测试集为 8:2 时，可以得到逻辑回归 AUC=0.846。

```
#梯度提升决策分类Gradient Boosting Classifier
from sklearn.ensemble import GradientBoostingClassifier
model = GradientBoostingClassifier()
model.fit( train_X , train_y )
```



(a) (b)

图 6　结果可视化

3.5 KNN

3.5.1 模型原理

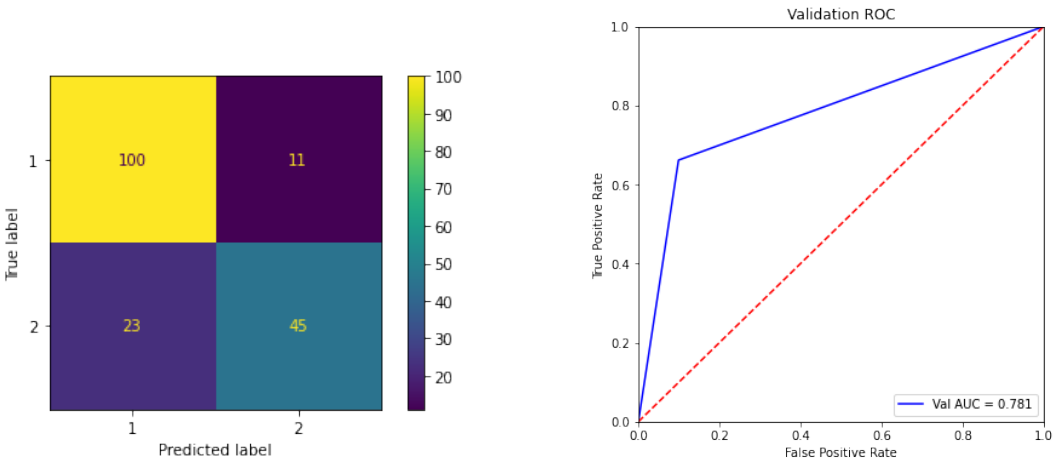
KNN 分类算法包括以下 4 个步骤:

- 准备数据，对数据进行预处理。
- 计算测试样本点（也就是待分类点）到其他每个样本点的距离。
- 对每个距离进行排序，然后选择出距离最小的 K 个点。
- 对 K 个点所属的类别进行比较，根据少数服从多数的原则，将测试样本点归入在 K 个点中占比最高的那一类

3.5.2 模型结果

经过运行，当训练集比测试集为 8:2 时，可以得到逻辑回归 AUC=0.781。

```
#KNN最邻近算法 K-nearest neighbors
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors = 3)
model.fit( train_X , train_y )
```



(a) (b)

图 7　结果可视化

3.6 朴素贝叶斯分类

3.6.1 模型原理

朴素贝叶斯分类（NBC）是以贝叶斯定理为基础并且假设特征条件之间相互独立的方法，先通过已给定的训练集，以特征词之间独立作为前提假设，学习从输入到输出的联合概率分布，再基于学习到的模型，输入 X 求出使得后验概率最大的输出 Y。

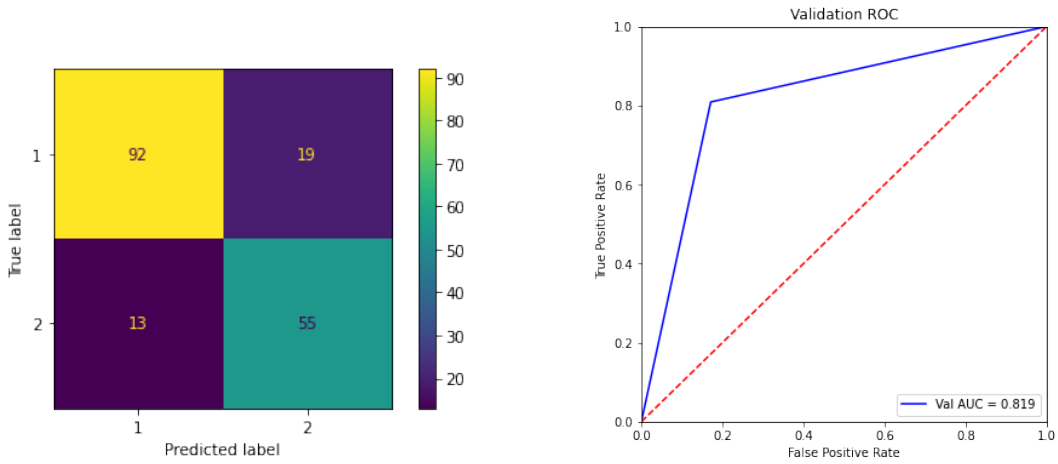
由于 $P(X)$ 的大小是固定不变的，因此在比较后验概率时，只比较上式的分子部分即可。因此可以得到一个样本数据属于类别 y_i 的朴素贝叶斯计算:

$$P\left(y_i \mid x_1, x_2, \cdots, x_d\right)=\frac{P\left(y_i\right) \prod_{j=1}^{d} P\left(x_j \mid y_i\right)}{\prod_{j=1}^{d} P\left(x_j\right)} \tag{5}$$

3.6.2 模型结果

经过运行，当训练集比测试集为 8:2 时，可以得到逻辑回归 AUC=0.819。

```
#朴素贝叶斯分类 Gaussian Naive Bayes
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit( train_X , train_y )
```



(a) (b)

图 8　结果可视化

3.7 模型评价

设置训练集数据与测试集数据范围为 0.6-0.9，计算各个模型的分数，可以得到下图：

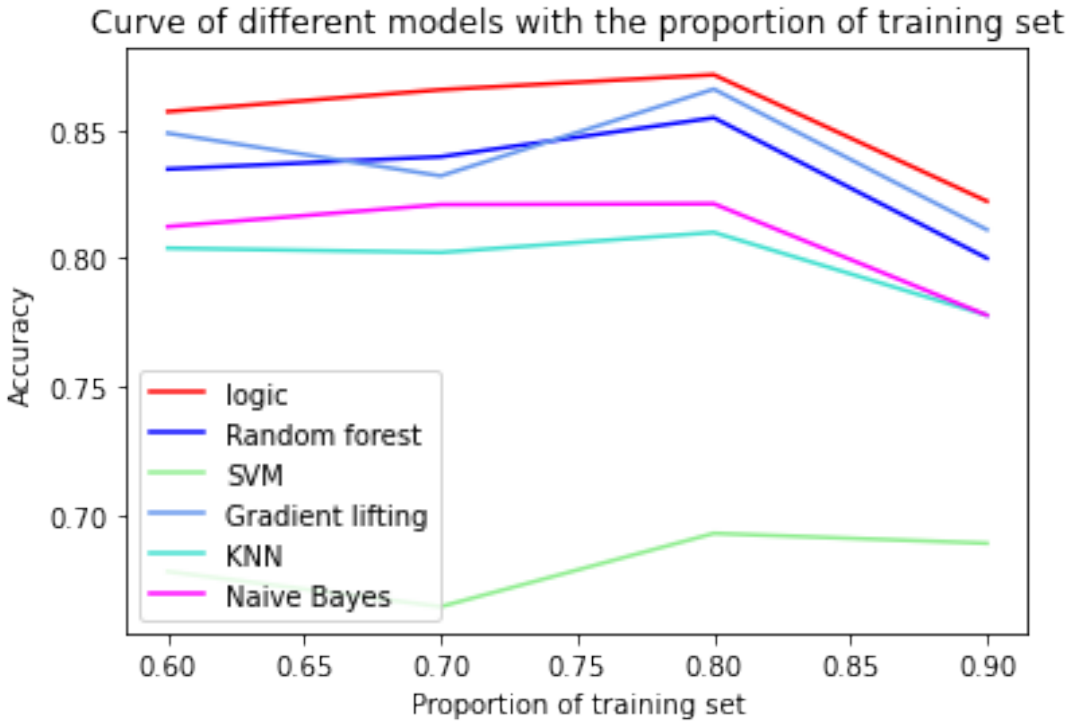


图 9　数据预处理

可以看到，效果最好的为逻辑回归模型。