杭州电子科技大学

《数据挖掘课程设计》

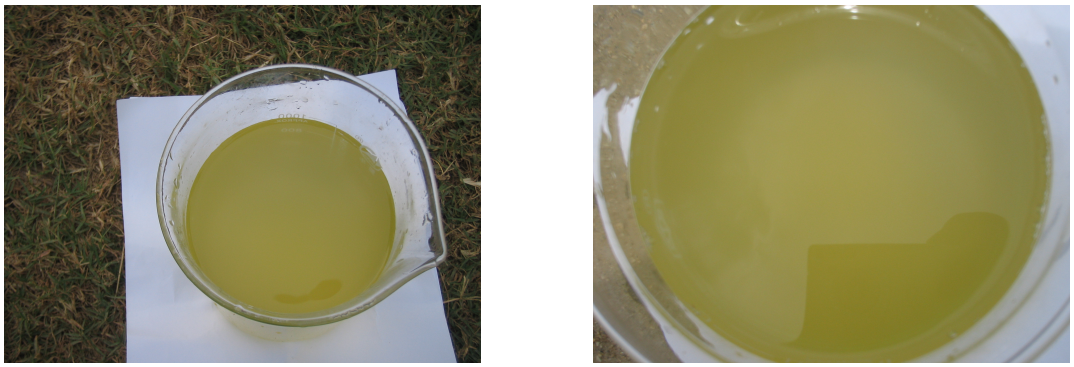
平时作业1

专　　业	信息与计算科学
班　　级	19073112
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 7月 4日

1. 水质图片分类

1.1 数据集介绍

该数据集为某湖水样本，共 203 张图片，由类别 + 下划线 + 序号命名。



(a) (b)

图 1　数据集

1.2 数据预处理

1.2.1 图片裁剪

为了提取有效信息，规范数据输入，只裁剪中间含有有水质样本的图像。保留中心 100×100 的图像。

图 2　裁剪后

1.2.2 计算颜色矩

获取裁剪的图像后，对其 RGB 三个通道分别计算一到三阶颜色矩。之后采用 str.split 提取图片的类别，并将其存入 Dataframe

1.3 基于 SVM 的水质图片分类模型

对于此类小样本，采用 SVM 模型对其进行训练。

首先将预处理好的数据分割为训练集与测试集。之后调用 sklearn.svm 进行模型训练。

1.4 模型结果

采取混淆矩阵对训练结果进行模型评价。可以看到 SVM 模型对 1,2,3 类水质预测较好。4,5 类为不平衡数据。

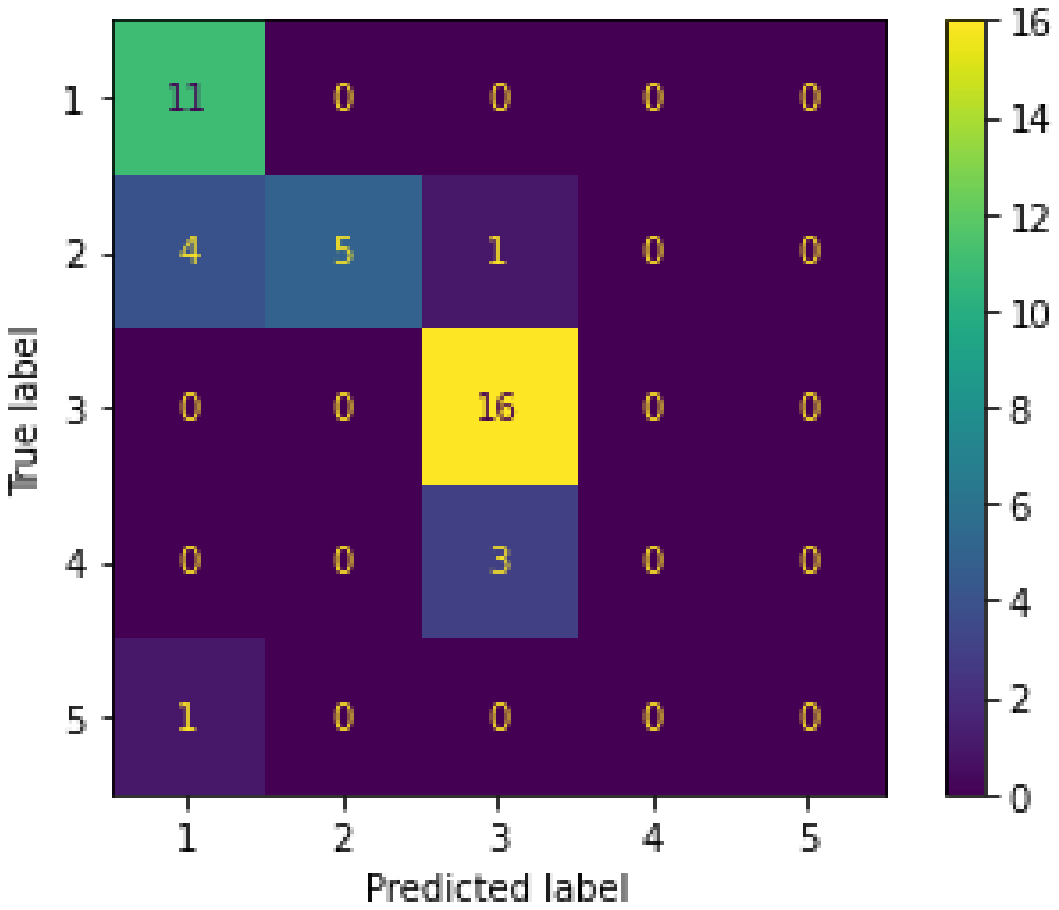


图 3　测试集混淆矩阵

杭州电子科技大学

《数据挖掘课程设计》

平时作业2

专　　业	信息与计算科学
班　　级	19073112
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 7月 4日

2. 航空公司客户价值分析

2.1 数据集介绍

该数据为某航空公司客户信息，包含会员档案信息和其乘坐航班记录等信息。

LOAD_TIME	FFP_DATE	LAST_TO_END	FLIGHT_COUNT	SEG_KM_SUM	AVG_DISCOUNT
2014/4/1	2006/03/31	6.6	3	18770	0.66
2014/4/1	2006/03/31	3.8	24	35087	0.62
2014/4/1	2006/04/07	2.8	9	20660	0.52
2014/4/1	2006/08/10	1	12	23071	0.51
2014/4/1	2008/02/07	3.17	3	2897	0.95
2014/4/1	2010/09/16	1.57	3	4608	0.65
2014/4/1	2011/04/28	17.83	2	3390	0.48
2014/4/1	2012/03/09	4.13	8	11797	1.35
2014/4/1	2012/08/31	5.9	6	6355	0.75
2014/4/1	2005/09/29	0.4	54	62170	0.79
2014/4/1	2009/01/23	2.33	24	15894	0.6
2014/4/1	2009/01/30	0.07	13	19517	0.72
2014/4/1	2009/01/30	4.63	10	12686	0.55
2014/4/1	2009/02/13	14.93	13	10992	1.33
2014/4/1	2009/04/25	10.27	3	4137	0.67
2014/4/1	2009/06/12	0.33	19	37415	0.63
2014/4/1	2010/03/25	1.63	13	24156	0.79
2014/4/1	2010/04/15	22.23	3	1559	0.87
2014/4/1	2010/05/20	23.17	2	1870	0.6
2014/4/1	2010/07/08	2.6	30	46621	0.93
2014/4/1	2011/03/10	4.57	4	7999	0.58

图 4　数据信息

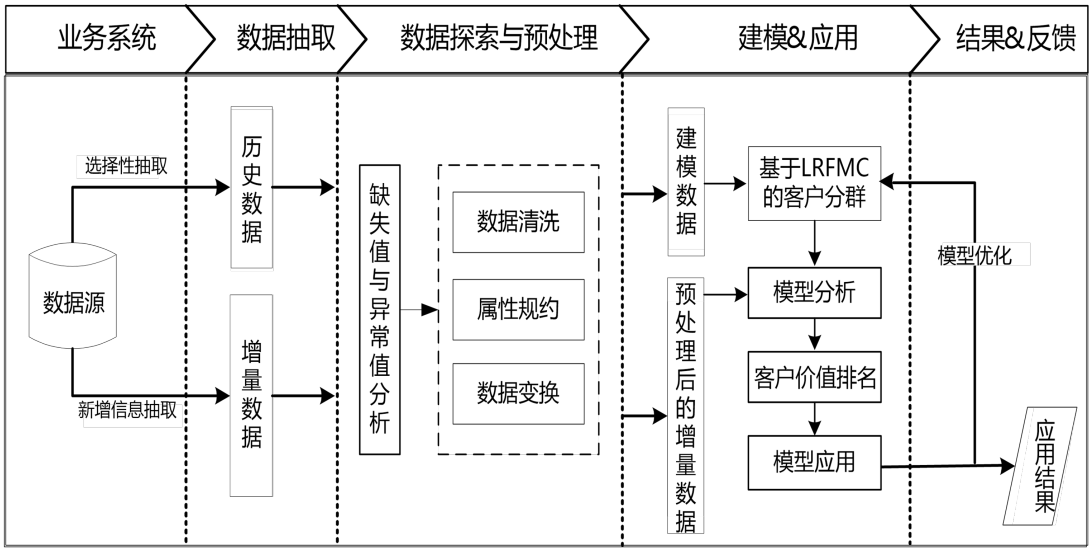


图 5　流程图

2.2 数据预处理

首先，以 2014-03-31 为结束时间，选取宽度为两年的时间段作为分析观测窗口，抽取观测窗口内有乘机记录的所有客户的详细数据形成历史数据。对于后续新增的客户详细信息，利用其数据中最大的某个时间点作为结束时间，采用上述同样的方法进行抽取，形成增量数据。根据末次飞行日期，从航空公司系统内抽取 2012-04-01 至 2014-03-31 内所有乘客的详细数据，总共 62988 条记录

原始数据中存在票价为空值，票价为空值的数据可能是客户不存在乘机记录造成。票价最小值为 0、折扣率最小值为 0、总飞行公里数大于 0 的数据。其可能是客户乘坐 0 折机票或者积分兑换造成。

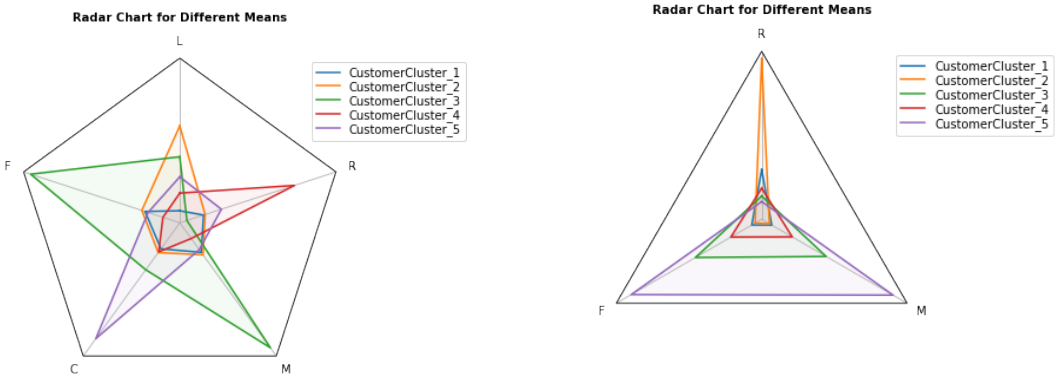
原始数据中属性太多，根据 LRFMC 模型，选择与其相关的六个属性，删除不相关、弱相关或冗余的属性。

2.3 基于 K-means 的客户聚类

采用 K-Means 聚类算法对客户数据进行分群，将其聚成五类

2.4 模型结果

对聚类结果进行特征分析，其中客户群 1 在 F、M 属性最大，在 R 属性最小；客户群 2 在 L 属性上最大；客户群 3 在 R 属性上最大，在 F、M 属性最小；客户群 4 在 L、C 属性上最小；客户群 5 在 C 属性上最大。



(a) (b)

图 6　结果可视化

杭州电子科技大学

《数据挖掘课程设计》

平时作业3

专　　业	信息与计算科学
班　　级	19073112
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 7月 4日

3. 基于用户的协同过滤

3.1 数据集说明

该数据包括 IMDB 用户对不同电影的打分情况，分为用户表，电影表，用户看过的电影及打分情况表。其中，打分情况表共 100000 行。

3.2 相关系数的计算

实现基于用户的协同过滤算法第一个重要的步骤就是计算用户之间的相似度。而计算相似度，建立相关系数矩阵目前主要分为以下几种方法。

3.2.1 皮尔森相关系数

皮尔逊相关系数一般用于计算两个定距变量间联系的紧密程度，它的取值在 [-1，+1] 之间。用数学公式表示，皮尔森相关系数等于两个变量的协方差除于两个变量的标准差。计算公式如下所示：

$$s(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

由于皮尔逊相关系数描述的是两组数据变化移动的趋势，所以在基于用户的协同过滤系统中，经常使用。描述用户购买或评分变化的趋势，若趋势相近则皮尔逊系数趋近于 1，也就是我们认为相似的用户。

3.2.2 基于欧几里德距离的相似度余弦相似度

欧几里德距离计算相似度是所有相似度计算里面最简单、最易理解的方法。计算出来的欧几里德距离是一个大于 0 的数，为了使其更能体现用户之间的相似度，可以把它规约到 (0,1] 之间，最终得到如下计算公式

$$s(X,Y) = \frac{1}{1 + \sum \sqrt{(X_i - Y_i)^2}} \tag{2}$$

只要至少有一个共同评分项，就能用欧几里德距离计算相似度；如果没有共同评分项，那么欧几里德距离也就失去了作用。其实照常理理解，如果没有共同评分项，那么意味着这两个用户或物品根本不相似。

3.2.3 余弦相似度

余弦相似度用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小。余弦相似度更加注重两个向量在方向上的差异，而非距离或长度上。计算公式如下所

示：

$$s(X,Y)=\cos\theta=\frac{\vec{x}*\vec{y}}{\|x\|*\|y\|} \tag{3}$$

3.3 基于用户的协同过滤

基于用户的协同过滤算法，另一个重要的步骤就是计算用户 u 对未评分商品的预测分值。首先根据上一步中的相似度计算，寻找用户 u 的邻居集 N□U, 其中 N 表示邻居集，U 表示用户集。然后，结合用户评分数据集，预测用户 u 对项 i 的评分，计算公式如下所示:

$$p_{u,i}=\bar{r}+\frac{\sum_{u'\subset N}s\left(u-u'\right)\left(r_{u',i}-\bar{r}_{u'}\right)}{\sqrt{\sum_{u'\subset N}\left|s\left(u-u'\right)\right|}} \tag{4}$$

其中，s(u- u') 表示用户 u 和用户 u' 的相似度。

3.4 算法的实现

现有的部分电影评分数据如下表:

	UserId	MovieId	Rank	No
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596

图 7　部分数据

实现代码如下所示:

```
def similarity(userid1:int,userid2:int,data:pd.DataFrame):
  # 获取userid1的电影id 并将其作为索引
  movie_index = list(data['MovieId'][data['UserId']==userid1].value_counts().index)

  #创建dataframe
  join_dataframe = pd.DataFrame(index=movie_index)

  #获取数据
  data1 = data[['MovieId','Rank']][data['UserId']==userid1].set_index('MovieId')
  data2 = data[['MovieId','Rank']][data['UserId']==userid2].set_index('MovieId')
```

```
    #换列名
    data1.columns = ['userid1']
    data2.columns = ['userid2']

    #join 连接
    join_dataframe = join_dataframe.join(data1)
    join_dataframe = join_dataframe.join(data2)
    join_dataframe = join_dataframe.fillna(0)

    # 计算相关系数
    norm1=np.sqrt(np.dot(join_dataframe['userid1'],join_dataframe['userid1']))
    norm2=np.sqrt(np.dot(join_dataframe['userid2'],join_dataframe['userid2']))
    val=np.dot(join_dataframe['userid1'],join_dataframe['userid2'])/(norm1*norm2)

    return val

c_user = 93
s = {}
all_user_id = data['UserId'].unique()

for cid in all_user_id:
s[cid] = similarity(cid,c_user,data)
s[c_user] = 0
Max = max(s.items(),key=lambda x:x[1] if x[1]<=1 else 0) # items以列表返回可遍历的(键, 值)
    元组数组

Max = max(s.items(),key=lambda x:x[1] if x[1]<=1 else 0) # items以列表返回可遍历的(键, 值)
    元组数组
```

3.5 模型结果

图 8

可以看到，与用户 20 最相似的为 769，相似度为 0.59，因此可以对 20 推荐 14, 125, 151, 275, 276, 283, 412, 477, 815, 820, 845, 866 电影。

杭州电子科技大学

《数据挖掘课程设计》

平时作业4

专　　业	信息与计算科学
班　　级	19073112
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 7月 4日

4. 中医证型的关联规则挖掘

4.1 数据集说明

该数据集为某机构收集的中医证型与乳腺癌阶段表，主要变量为六种证型 (肝气郁结, 热毒蕴结, 冲任失调，气血两虚，脾胃虚弱，肝肾阴虚) 与 TNM 分期。

4.2 数据预处理

为了建模需要，需要对数据进行离散化。本例采用分箱对各个证型系数进行离散化处理，将每个属性聚成四类。

实现代码如下所示：

```
#数据预处理
columns = data.columns
key = list(columns)[0:6]
values = ['A','B','C','D','E','F']
dict(zip(key,values))
for i in range(6):
   cutbin = np.linspace(data[key[i]].min(),data[key[i]].max(),5)
   label = [values[i]+str(j) for j in range(1,5)]
   data[key[i]] = pd.cut(data[key[i]],bins=cutbin,labels=label)
```

4.3 基于 Apriori 的关联规则挖掘

处理后的数据如下：

	肝气郁结证型系数	热毒蕴结证型系数	冲任失调证型系数	气血两虚证型系数	脾胃虚弱证型系数	肝肾阴虚证型系数	病程阶段	TNM分期	转移部位	确诊后几年发现转移
0	A1	B3	C2	D3	E1	F3	S4	H4	R1	J1
1	A4	B1	C2	D3	E1	F2	S4	H4	R1	J1
2	A1	B1	C2	D1	E1	F1	S4	H4	R2	J2
3	A3	B2	C2	D1	E2	F3	S4	H4	R2	J1
4	A2	B2	C1	D2	E2	F3	S4	H4	R2R5	J1
...	...	...	...	...	...	...	...	...	...	...
925	A3	B1	C2	D1	E1	F1	S1	H1	R0	J0
926	A1	B2	C1	D1	E2	F2	S1	H1	R0	J0
927	A3	B1	C2	D1	E1	F2	S1	H1	R0	J0
928	A3	B2	C1	D2	E1	F2	S1	H1	R0	J0
929	A2	B2	C1	D1	E2	F2	S1	H1	R0	J0

图 9

4.4 模型结果与分析

设定最小支持度为 0.07，最小置信度为 0.9，结果为

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

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

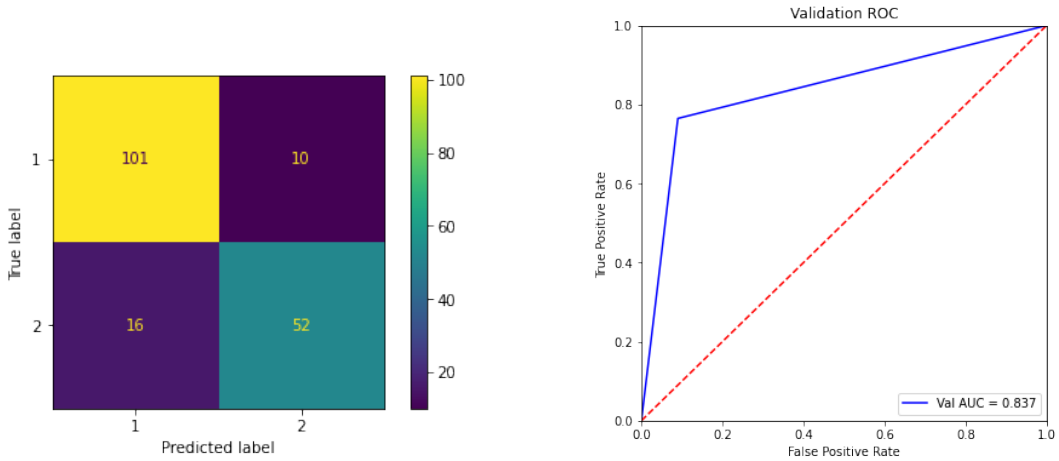
其中 $s(\cdot)$ 称为 Sigmoid 函数

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

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。

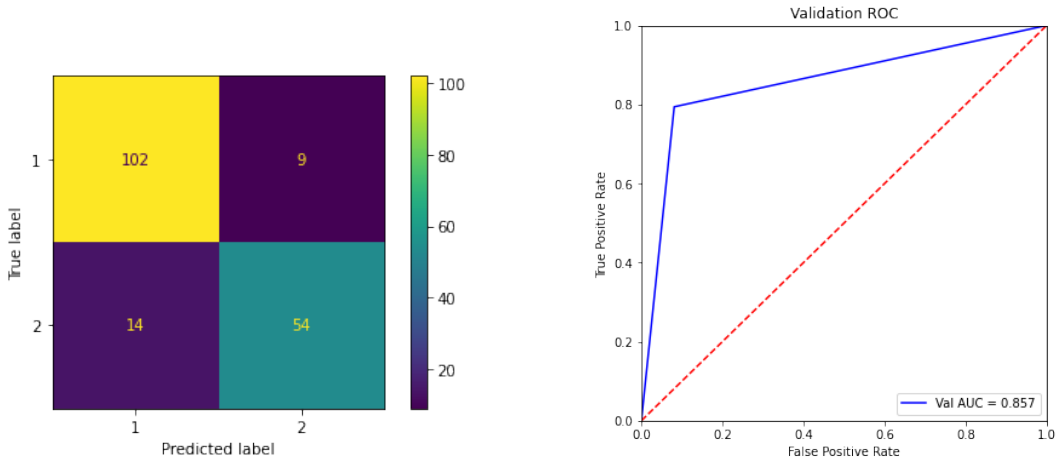
（2）令训练样例的输入特征的个数为 M，m 远远小于 M，则我们在每颗决策树的每个节点上进行分裂时，从 M 个输入特征里随机选择 m 个输入特征，然后从这 m 个输入特征里选择一个最好的进行分裂。m 在构建决策树的过程中不会改变。这里注意，要为每个节点随机选出 m 个特征，然后选择最好的那个特征来分裂。

（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

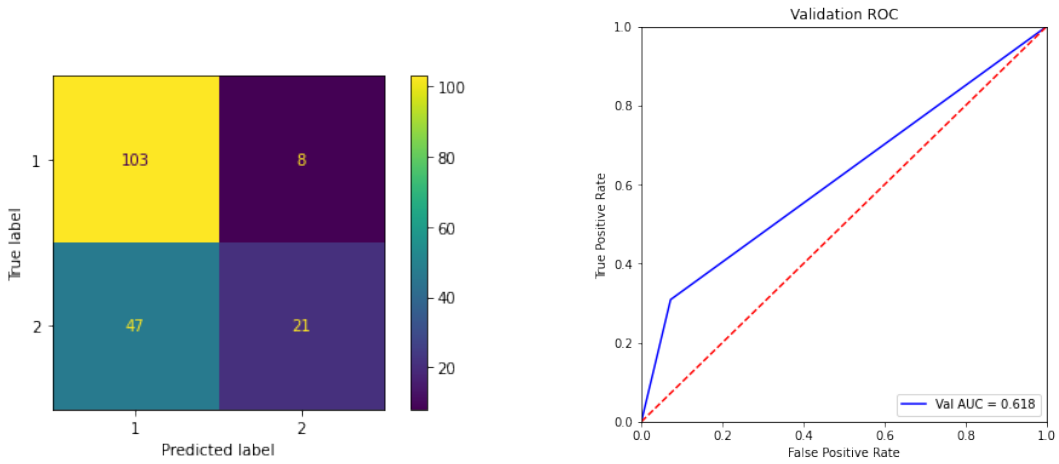
3.3.1 模型原理

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

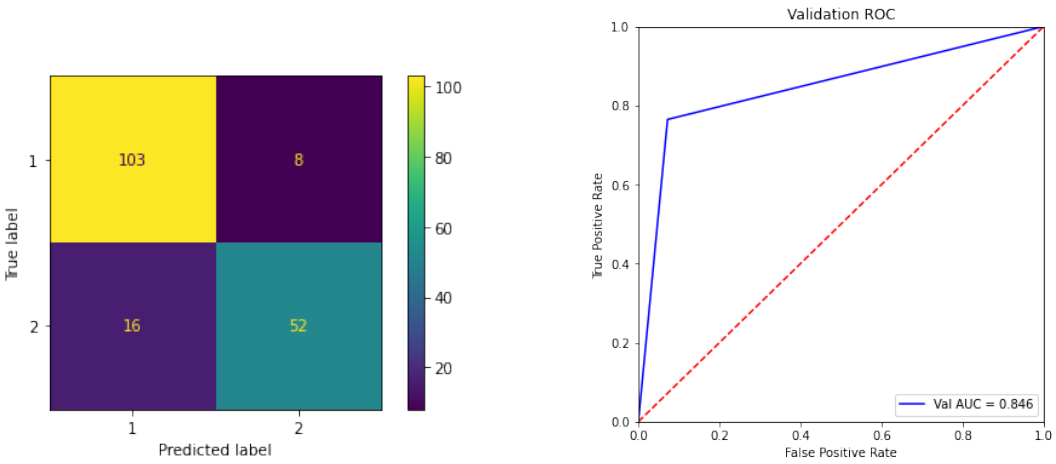
GBDT 是通过采用加法模型（即基函数的线性组合），以及不断减小训练过程产生的残差来达到将数据分类或者回归的算法，Friedman 提出了利用最速下降的近似方法，利用利用损失函数的负梯度在当前模型的值，作为回归问题中提升树算法的残差的近似值，拟合一个回归树。

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

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

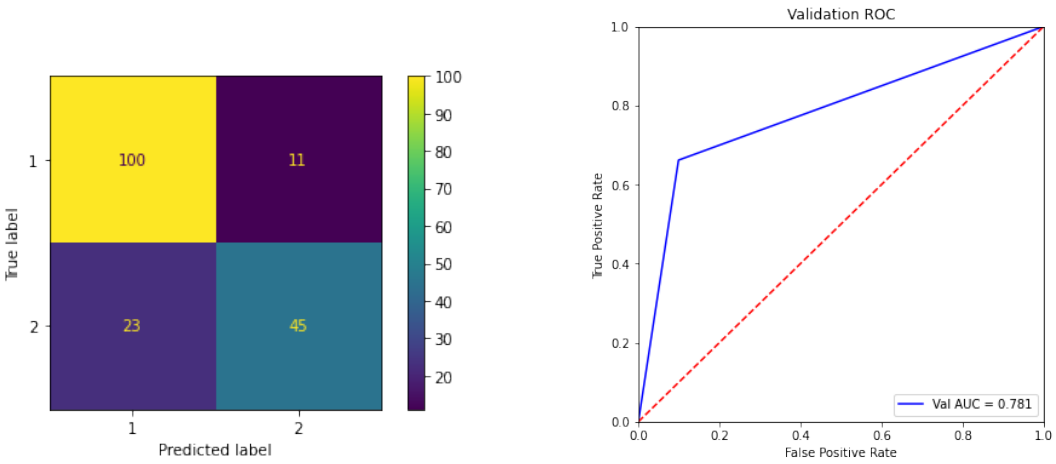
KNN 分类算法包括以下 4 个步骤:

- 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。

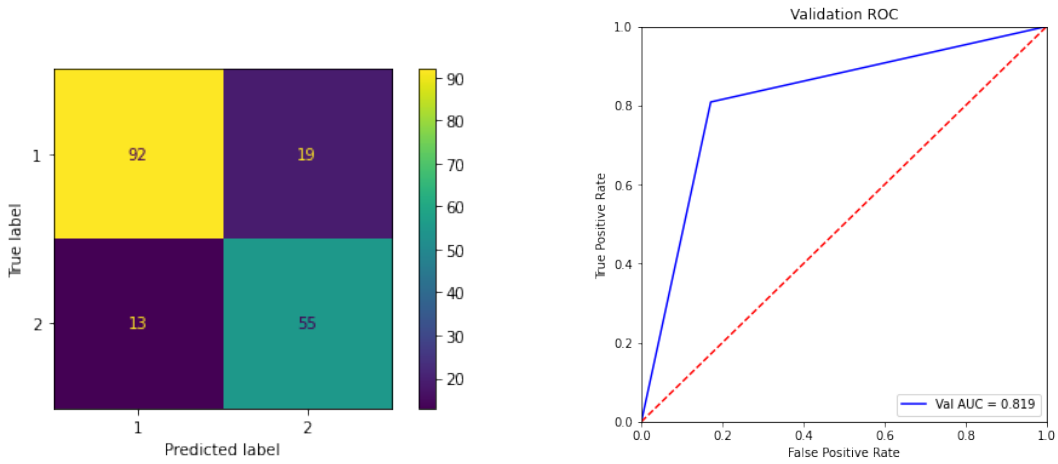
由于 $P(X)$ 的大小是固定不变的，因此在比较后验概率时，只比较上式的分子部分即可。因此可以得到一个样本数据属于类别 y_i 的朴素贝叶斯计算:

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

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：

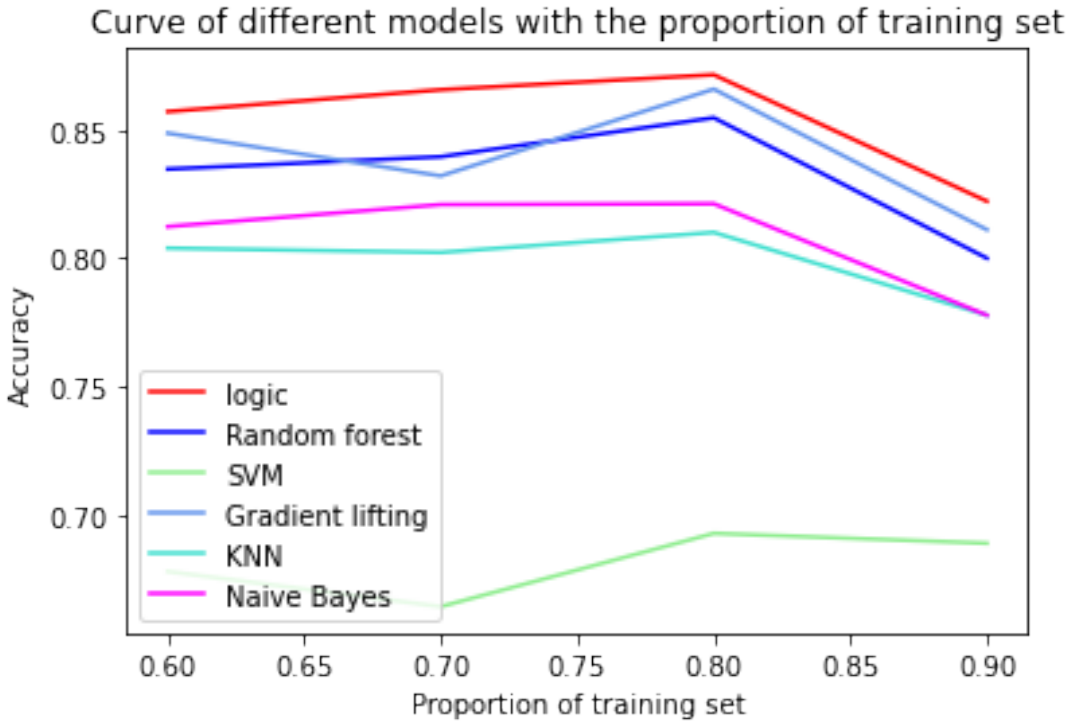


图 9　数据预处理

可以看到，效果最好的为逻辑回归模型。

1: (’A1’,): 111, (’S4’,): 255, (’H4’,): 415, (’J1’,): 260, (’R2’,): 80, (’J2’,): 75, (’A3’,): 275, (’A2’,): 508, (’J3’,): 80, (’S3’,): 165, (’R0’,): 515, (’J0’,): 510, (’S2’,): 340, (’S1’,): 170, (’H3’,): 205, (’H2’,): 205, (’H1’,): 105,

2: (’A1’, ’J0’): 85, (’A1’, ’R0’): 85, (’A2’, ’H2’): 114, (’A2’, ’H3’): 141, (’A2’, ’H4’): 229, (’A2’, ’J0’): 284, (’A2’, ’J1’): 129, (’A2’, ’R0’): 284, (’A2’, ’S2’): 188, (’A2’, ’S3’): 118, (’A2’, ’S4’): 137, (’A3’, ’H4’): 143, (’A3’, ’J0’): 122, (’A3’, ’J1’): 100, (’A3’, ’R0’): 127, (’A3’, ’S1’): 72, (’A3’, ’S2’): 96, (’A3’, ’S4’): 71, (’H1’, ’J0’): 105, (’H1’, ’R0’): 105, (’H2’, ’J0’): 200, (’H2’, ’R0’): 205, (’H2’, ’S4’): 75, (’H3’, ’J0’): 200, (’H3’, ’R0’): 200, (’H3’, ’S4’): 70, (’H4’, ’J1’): 255, (’H4’, ’J2’): 75, (’H4’, ’J3’): 80, (’H4’, ’R2’): 80, (’H4’, ’S2’): 205, (’H4’, ’S3’): 100, (’H4’, ’S4’): 80, (’J0’, ’R0’): 510, (’J0’, ’S1’): 135, (’J0’, ’S2’): 130, (’J0’, ’S3’): 70, (’J0’, ’S4’): 175, (’J1’, ’S2’): 105, (’J1’, ’S3’): 80, (’R0’, ’S1’): 140, (’R0’, ’S2’): 130, (’R0’, ’S3’): 70, (’R0’, ’S4’): 175,
W 3: (’A1’, ’J0’, ’R0’): 85, (’A2’, ’H2’, ’J0’): 114, (’A2’, ’H2’, ’R0’): 114, (’A2’, ’H3’, ’J0’): 141, (’A2’, ’H3’, ’R0’): 141, (’A2’, ’H4’, ’J1’): 129, (’A2’, ’H4’, ’S2’): 118, (’A2’, ’J0’, ’R0’): 284, (’A2’, ’J0’, ’S2’): 70, (’A2’, ’J0’, ’S4’): 103, (’A2’, ’R0’, ’S2’): 70, (’A2’, ’R0’, ’S4’): 103, (’A3’, ’H4’, ’J1’): 95, (’A3’, ’J0’, ’R0’): 122, (’H1’, ’J0’, ’R0’): 105, (’H2’, ’J0’, ’R0’): 200, (’H2’, ’J0’, ’S4’): 75, (’H2’, ’R0’, ’S4’): 75, (’H3’, ’J0’, ’R0’): 200, (’H3’, ’J0’, ’S4’): 70, (’H3’, ’R0’, ’S4’): 70, (’H4’, ’J1’, ’S2’): 100, (’H4’, ’J1’, ’S3’): 80, (’J0’, ’R0’, ’S1’): 135, (’J0’, ’R0’, ’S2’): 130, (’J0’, ’R0’, ’S3’): 70, (’J0’, ’R0’, ’S4’): 175,

4: (’A2’, ’H2’, ’J0’, ’R0’): 114, (’A2’, ’H3’, ’J0’, ’R0’): 141, (’A2’, ’J0’, ’R0’, ’S2’): 70, (’A2’, ’J0’, ’R0’, ’S4’): 103, (’H2’, ’J0’, ’R0’, ’S4’): 75, (’H3’, ’J0’, ’R0’, ’S4’): 70