

华东理工大学

模式识别大作业

题目	<u>泰坦尼克比赛</u>
院系	<u>信息科学与工程学院</u>
专业	<u>控制科学与工程</u>
组员	<u>吴骏逸叶朋飞朱雯文金字尘赵帅</u>
指导老师	<u>赵海涛</u>

泰坦尼克比赛

组员：吴骏逸朱雯文叶朋飞金宇尘赵帅

一、泰坦尼克大赛简介

沉没的泰坦尼克号是历史上最臭名昭著的沉船。1912年4月15日，在处女航时，泰坦尼克号撞上冰山后沉没，2224名乘客和机组人员1502人死亡。这一轰动性的悲剧震惊了国际社会，并促使诞生了更完善的船舶安全条例。海难造成大量生命死亡的原因是没有足够的救生艇的乘客和船员。虽然有一些运气的因素，在下沉过程中，一些群体的人更可能生存比别人，如妇女，儿童和上层阶级。

在这个挑战中，我们要求完成对什么样的人可能生存的分析。特别是，我们要求使用机器学习的工具来预测乘客在悲剧中幸存下来。

二、整体解决方案

将船上游客的生死抽象为一个0-1分类问题，在经过组员间的讨论与对原始数据的研究的基础上，我们首先确定了一个大的求解的框架，就是把这个问题看成一个分类问题，利用数据，找出影响生存的因素，并且对这些“因素”与“是否生存”这两者关系进行建模。

我们通过建立模型后，在训练数据上反复调试模型中的部分参数，使得分类器对于训练数据达到较好的分类效果后，将该模型用于测试数据的预测，并分析其预测效果。

2.1 数据读入

● 数据可从Kaggle下载（需要先注册）。数据包括训练数据train.csv，和测试数据test.csv。内容如图所示：

1	Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, M	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, I	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikinen, I	female	26	0	0	STON/O2.	7.925		S
5	4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr.	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy, male		54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, M	male	2	3	1	349909	21.075		S
10	9	1	3	Johnson, N	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, M	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, M	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saunders	male	20	0	0	A/5. 2151	8.05		S

图1 测试样本信息

其中测试数据一共有 891 组，训练数据有 418 组。数据所表示的意思如下所示：

- PassengerId//游客 id
- Survived//1 幸存
- Pclass//1 2 3 表示社会地位
- Name//姓名
- Sex//性别
- Age//年龄
- SibSp//同船上的兄弟、姐妹数量
- Parch//父母子女数量
- Ticket//船票号码
- Fare//花费金额
- Cabin//船舱号码
- Embarked//S C Q 上船地点

```
Train = readtable('train.csv','Format','%f%f%f%q%s%f%f%f%q%f%q%s');  
Test = readtable('test.csv','Format','%f%f%q%s%f%f%f%q%f%q%s');  
disp(Train(1:5,[2:3 5:8 10:11]))
```

注：matlab 版本越高越好，如为 Matlab2015 后的版本，可替换%s 为%C，如果都不能运行，需重新安装高版本 matlab，或用其他语言编程，如 python。

● 按性别预测生还的可能性

```
disp(grpstats(Train(:,{'Survived','Sex'}), 'Sex'))
```

分类器为女生能够生还，男生不能生还。这个分类器给出的生还结果可以作为 baseline，识别率可如下命令得到。

```
gendermdl = grpstats(Train(:,{'Survived','Sex'}), {'Survived','Sex'});  
all_female = (gendermdl.GroupCount('0_male') +  
gendermdl.GroupCount('1_female'))...
```

/

```
sum(gendermdl.GroupCount)
```

2.2 数据预处理

- 数据中可能存在数据缺失，数据不对的情况，以及很大的噪声等情况。

当有数据缺失的记录在整个数据中只占一个很小比例时，可以直接删除缺失记录，对余下的完全数据进行处理。但是在实际数据中，往往缺失数据占有相当的比重，这样做不仅会产生偏差，甚至会得出有误导性的结论，同时丢失大量信息，造成浪费。因此我们使用一种新的方法来进行处理。填补法是处理数据缺失时普遍使用的一种技术，就是说给各个缺失数据找一个填充值，用这样的方法得到“完整数据”，然后用标准正常的完整数据的统计方法进行数据分析和推断。包括平均值填充法、热卡填充法（或就近补齐）、使用任何可能的值填充等方法。

- train 数据中的 Cabin 中缺失值比较多，另外 Fare=0 是比较奇怪的情况。

```
Train.Fare(Train.Fare == 0) = NaN; % 票价为 0 的项设置为 NaN
Test.Fare(Test.Fare == 0) = NaN; % 票价为 0 的项设置为 NaN
vars = Train.Properties.VariableNames; % 提取所有特征列的名字
figure
imagesc(ismissing(Train))
set(gca,'XTick', 1:12,'XTickLabel',vars);
```

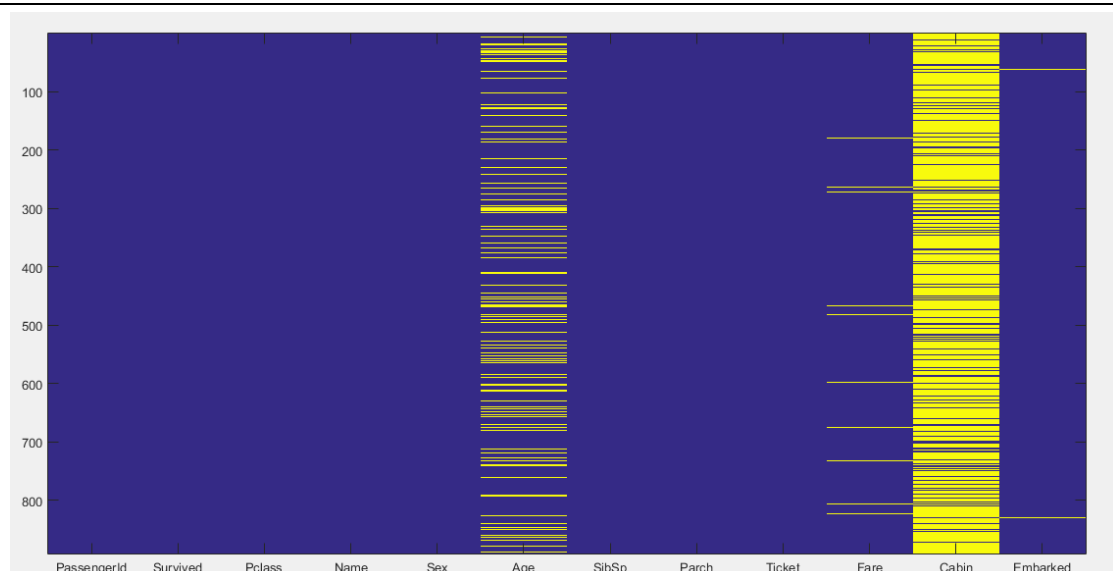


图 2 特征缺失值展现图

- 平均值填充法

在处理数据时可以把变量分为数值型和非数值型。如果是非数值型的缺失数据，运用统计学中众数的原理，用此变量在其他对象中取值频数最多的值来填充缺失值；如果是数值型的缺失值，则取此变量在其他所有对象的取值均值来补齐缺失值。

采用此方法，用平均年龄替代没有年龄的

```
avgAge = nanmean(Train.Age) % 计算平均年龄
Train.Age(isnan(Train.Age)) = avgAge; % 用平均值代替 NaN
Test.Age(isnan(Test.Age)) = avgAge;
```

● 热卡填充法（或就近补齐）

对于一个包含空值的变量，本方法是在完整数据中找到一个与空值最相似的变量，然后用这个相似的值来进行填充。与均值替换法相比，本方法简单易懂还可以保持数据本身的类型，利用本方法填充数据后，其变量值与填充前很接近。但是这种方法也存在不足之处，就是其主观因素较多，还比较耗时。

费用按舱位的级别填入各自的均值

```
fare = grpstats(Train(:,{'Pclass','Fare'}),'Pclass'); % 计算舱位的平均值
disp(fare)
for i = 1:height(fare) % 对每个舱位
    % 用平均值代替缺失值
    Train.Fare(Train.Pclass == i&isnan(Train.Fare)) = fare.mean_Fare(i);
    Test.Fare(Test.Pclass == i&isnan(Test.Fare)) = fare.mean_Fare(i);
end
```

● 其他有关 Cabin 的问题，多个 Cabin，无 Cabin，Cabin 不正常

```
train_cabins = cellfun(@strsplit, Train.Cabin, 'UniformOutput', false);
test_cabins = cellfun(@strsplit, Test.Cabin, 'UniformOutput', false);
% 计算船舱号码的数目
Train.nCabins = cellfun(@length, train_cabins);
Test.nCabins = cellfun(@length, test_cabins);
% 处理异常值-只有一等舱的人才有多多个船舱号码
Train.nCabins(Train.Pclass ~= 1 & Train.nCabins > 1,:) = 1;
```

```
Test.nCabins(Test.Pclass ~= 1 & Test.nCabins > 1,:) = 1;
% 如果数据缺失，则应为 0
Train.nCabins(cellfun(@isempty, Train.Cabin)) = 0;
Test.nCabins(cellfun(@isempty, Test.Cabin)) = 0;
```

● 使用任何可能的值填充

这种方法是用缺失值所有可能的数值来填充，能够起到一个补齐效果。而这种方法的缺点是，当要研究的数据量很大或者缺失的数值较多时，计算量很大，需要测试的方案很多。针对其缺点有另外一种方法，用一样的方法来填补缺失数，不同的是从结果相同的对象中选择所有可能情况的数值，而不是根据所有情况的对象进行尝试，这样能够在一定程度上缓解原方法的不足。

乘客没有登船地点，填入登船最多的地点

```
% 提取最高频率的值
disp(grpstats(Train(:, {'Survived', 'Embarked'}), 'Embarked'))
% 用出现频率最高的值代替缺失值
for i = 1 : 891
    if isempty(Train.Embarked{i})
        Train.Embarked{i} = 'S';
    end
end
for i = 1 : 418
    if isempty(Test.Embarked{i})
        Test.Embarked{i} = 'S';
    end
end
% 将数据类型从绝对性转换成 double
Train.Embarked = double(cell2mat(Train.Embarked));
Test.Embarked = double(cell2mat(Test.Embarked));
```

● Sex 变成 double 型

```
for i = 1 : 891
```

```

        ifstrcmp(Train.Sex{i} , 'male')
            Train.Sex{i}=1;
        else
            Train.Sex{i}=0;
        end
    end
end
for i = 1 : 418
    ifstrcmp(Test.Sex{i} , 'male')
        Test.Sex{i}=1;
    else
        Test.Sex{i}=0;
    end
end

Train.Sex = cell2mat(Train.Sex);
Test.Sex = cell2mat(Test.Sex);

```

- 去除不用的值，如姓名等，或缺失值太多

```

Train(:, {'Name', 'Ticket', 'Cabin'}) = [];
Test(:, {'Name', 'Ticket', 'Cabin'}) = [];

```

2.3 数据分析与可视化

缺失值填充之后，就要对其他格式有问题的属性进行处理了。比如 `Sex` `Embarked` 这些属性的值都是字符串类型的，而 `scikit learn` 中的模型都只能处理数值型的数据，需要将这些原始的字符串类型的数据转为数值型数据。所有数据通常可以分成两种类型：定量与定性。定量的属性（数值属性）通常蕴涵着可排序性，比如在泰坦尼克号数据集中，年龄就是一个定量属性。定性属性（标称序数二元属性）的值是一些符号或事务的名称，每个值代表某种类别编码或状态，不是可测量量，是不具有排序意义的，比如 `sex`。

在本次实验中，将 `sex` 变成 `double` 型。

```

fori = 1 : 891
    ifstrcmp(Train.Sex{i} , 'male')
        Train.Sex{i}=1;
    else
        Train.Sex{i}=0;
    end
end
fori = 1 : 418
    ifstrcmp(Test.Sex{i} , 'male')
        Test.Sex{i}=1;
    else
        Test.Sex{i}=0;
    end
end
Train.Sex = cell2mat(Train.Sex);
Test.Sex = cell2mat(Test.Sex);

```

● 对年龄进行可视化，直观分析生存者以及遇难者年龄的直方图，代码如下：

```

figure
hist (Train.Age(Train.Survived == 0)) % 遇难者年龄分布直方图
figure
hist (Train.Age(Train.Survived == 1)) % 幸存者年龄分布直方图

```

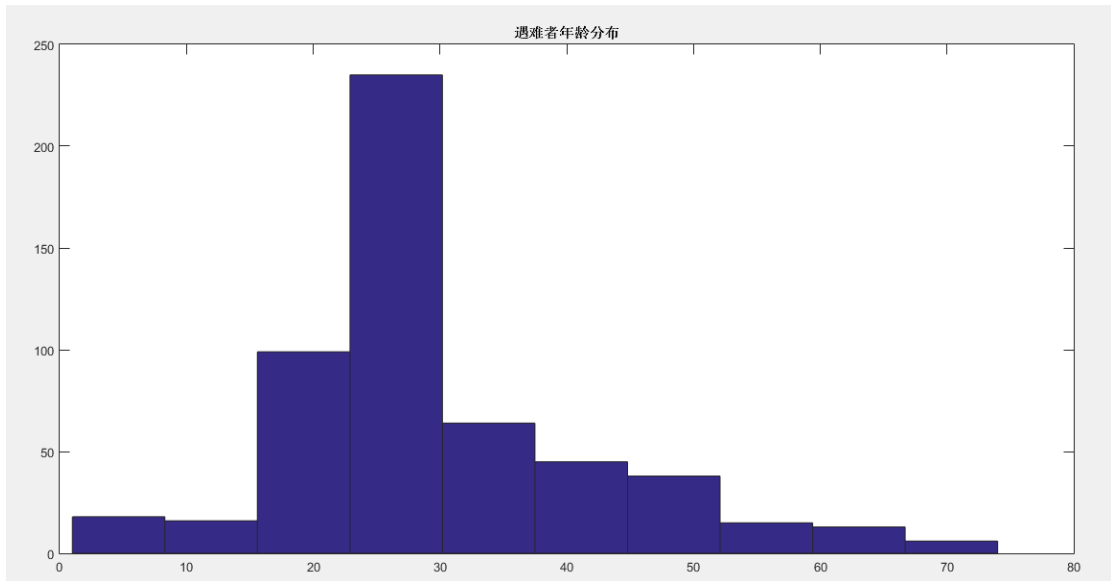



图 3 遇难者年龄分布图

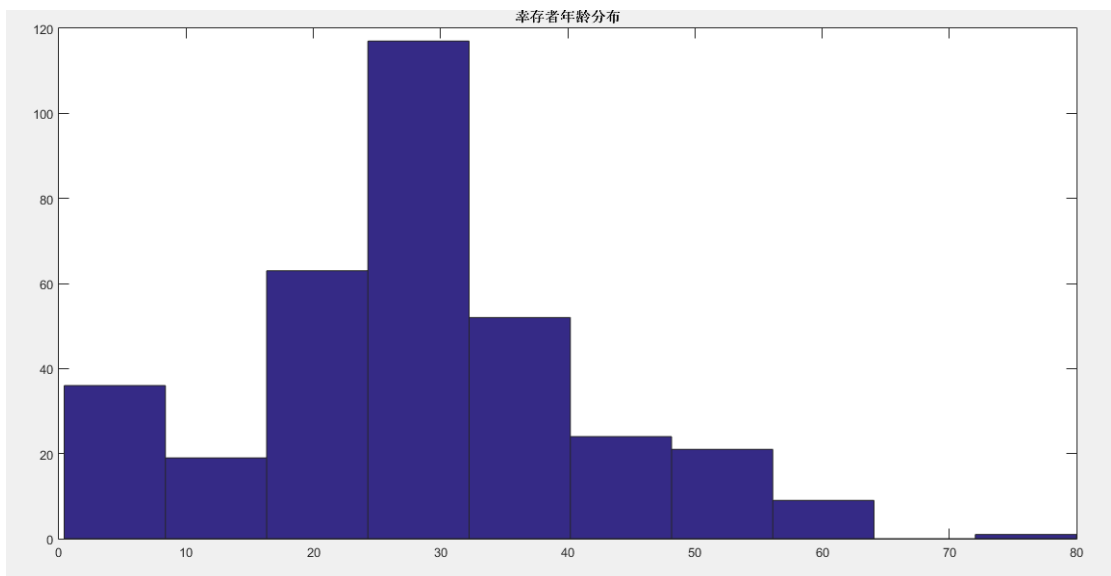


图 4 幸存者年龄分布图

2.4 特征值的选取

从线性空间的角度看，在一个定义了内积的线性空间里，对一个 N 阶对称方阵进行特征分解，就是产生了该空间的 N 个标准正交基，然后把矩阵投影到这 N 个基上。 N 个特征向量就是 N 个标准正交基，而特征值的模则代表矩阵在每个基上的投影长度。特征值越大，说明矩阵在对应的特征向量上的方差越大，功率越大，信息量越多。应用到数据挖掘中，意思就是最大特征值对应的特征向量方向上包含最多的信息量，如果某几个特征值很小，说明这几个方向信息量很

小，可以用来降维，也就是删除小特征值对应方向的数据，只保留大特征值方向对应的数据，这样做以后数据量减小，但有用信息量变化不大。

Name, Ticket, Cabin 这些字段太零散，基本上每个人的都不一样，感觉并没有什么用。Cabin 这一维度的特征更是缺失很严重，所以暂且不考虑 Ticket, Cabin 的这些特征。后面的随机森林法就是这么做的。

但是反观 Name 这个特征，看似并没有什么用，其实不然。通俗地说，Name 可以给模型提供一定的泛化能力，比如一家人面临危机的时候，大家肯定都先找到自己的家人一起逃生，所以一家人的存活状况相关性肯定很高的。所以我们小组经过讨论，决定引入名字特征的方式并不是直接引入名字，而是考虑和当前预测人的名字同姓的存活率。另外还有个背景问题，就是逃生的时候，女士优先逃生，这个时候家人就分开了，所以名字这个特征还要考虑性别，综合来说就是性别+姓的存活率作为一个特征。

按性别预测生还的可能性

分类器为女生能够生还，男生不能生还。这个分类器给出的生还结果可以作为 baseline，识别率可如下命令得到。

```
disp(grpstats(Train(:,{'Survived','Sex'}), 'Sex'))

gendermdl = grpstats(Train(:,{'Survived','Sex'}), {'Survived','Sex'});
all_female = (gendermdl.GroupCount('0_male') +
              gendermdl.GroupCount('1_female'))... / sum(gendermdl.GroupCount)
```

按 Name 中所带有的 Mrs 或 Miss 信息来判断某位女性是否结婚来预测生还的可能性

```
%% 判断是Miss还是Mrs还是Mr
for i=1:891
    if ~isempty(strfind(Train.Name{i}, 'Miss.'))
        Train.Name{i}=0;
    else
        if ~isempty(strfind(Train.Name{i}, 'Mrs.'))
```

```
Train.Name{i}=1;
else
Train.Name{i}=2;
end
end
end

for i=1:418
if ~isempty(strfind(Test.Name{i}, 'Miss.'))
Test.Name{i}=0;
else
if ~isempty(strfind(Test.Name{i}, 'Mrs.'))
Test.Name{i}=1;
else
Test.Name{i}=2;
end
end
end

Train.Name = cell2mat(Train.Name);
Test.Name = cell2mat(Test.Name);
```

另外，还应增加相应类别的存活率这些特征，比如各种性别的存活率以及各种等级的存活率，把分类 ID 扔进模型之后有必要把所属 ID 的百分比加进去，前者是“是什么类别”的因素，后者是“有多少存活比例”的因素，是和有不能混为一谈。

在进行数据处理时，不可避免的会遇到数据缺失，有如下一些处理方法：

- (1) 直接扔掉这行数据
- (2) 对于缺失的数据统一给一个新的 label，让模型来学出给这种 label

多大的权值

(3) 这个特征的缺失率很高

a) 直接扔掉这列特征

b) 搞一个模型来拟合这维度的特征

(4) 给一个默认值，这个值可以是均值，或者众数

(5) 使用回归随机森林等模型来预测缺失属性的值。因为年龄在该数据集里是一个相当重要的特征，所以保证一定的缺失值填充准确率是非常重要的，对结果也会产生较大影响。一般情况下，会使用数据完整的条目作为模型的训练集，以此来预测缺失值。对于当前的这个数据，可以使用随机森林来预测也可以使用线性回归预测。

2.5 实验方法及结论

(1) SVM

在此次分类过程中，我们看到这是 2 类的分类问题，但是是高维的，如果用线性分类器对其进行分类，这样就避免不了一部分数据会分错，因此我们将低维的数据映射到高维坐标系中，至于如何映射，我们采取的是核函数。在不断地尝试过程中，发现高斯径向基函数作为试验的核函数可以取得不错的效果。

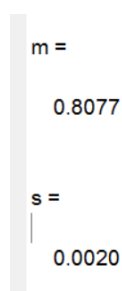
至于取的试验样本和测试样本，我们都是随机的，也就是说试验每次抽取 70% 训练样本以及剩下 30% 的测试用样本都是随机的，因此对分类结果有一定的影响，所以对于某一个给定核函数及其参数我们都是利用循环进行 50 次的试验，最后计算 50 次试验的评价正确率以及其方差。采样数据代码如下：

```
%% SVM算法
result=zeros(50,1)
for kk=1:50
data=table2array(Train(:,3:end));
    groups=table2array(Train(:,2)); %存活的人
    [train, test] = crossvalind('holdOut',groups,0.3);
cp = classperf(groups); %随机选择训练集合测试集
```

下面是关于 SVM 算法以及求正确率和方差的代码，如下：

```
svmStruct=svmtrain(data(train,:),groups(train),'Kernel_Function','rbf','rbf_sigma',2.2);
classes = svmclassify(svmStruct,data(test,:));
classperf(cp,classes,test);
result(kk,1)=cp.CorrectRate;
end
m=mean(result)
s=var(result)
```

经过多次试验，我们得到了试验的准确率为 80.77%，比较满意，方差也非常的小，可以看出解的结果比较稳定。



```
m =
    0.8077

s =
    0.0020
```

图 5 试验准确率及方差

(2) 随机森林法

```
Y_train = Train.Survived; % 加载目标值
%定义目标函数，该比赛的目的是为了估计泰坦尼克号成员的存活状况，所以是否存活自然是最终目标，并且只分两类，存活为 1，死亡为 0；
X_train = Train(:,3:end);% 选择预测变量
%将数据导入，并去除姓名、票号和船舱座位号等信息，将舱位等级、性别、年龄、亲人数量、票价以及登录地点作为变量
vars = X_train.Properties.VariableNames; %提取变量名
%将所给数据变量名导入，方便之后按照这些变量绘图标名进而分类
```

```

X_train = table2array(X_train); % 转换成数据矩阵

%将表格 X_train 中的数据转换成矩阵，同时，同一特征数据放在同一列，
matlab 是用矩阵运算的，从而构成训练样本数据矩阵

X_test = table2array(Test(:,2:end)); %转换成数据矩阵

%将表格 X_test 中的数据转换成矩阵，同时，同一特征数据放在同一列，
matlab 是用矩阵运算的，从而构成测试数据矩阵

categoricalPredictors = {'Pclass', 'Sex', 'Embarked'};

%将舱位等级、性别和登船地点作为之后进行预测的参考变量

rng(1); % 再现性

%通过 RAND 等产生一系列随机数，将之作为各个变量的初始权重系数

c = cvpartition(Y_train,'holdout', 0.30); % 取出 30%的数据用作交叉检验

%随机划分训练样本中 30%的数据作为测试数据进行交叉检验

RF = TreeBagger(200, X_train(training(c),:), Y_train(training(c)),...
'PredictorNames', vars, 'Method','classification',...
'CategoricalPredictors', categoricalPredictors, 'oobvarimp', 'on');

%调用随机森林函数，建立 200 个小的分类器对训练数据进行分类，随机森
林函数通过训练样本确定的参数，然后将 200 个小分类器综合成一个大的分类
器，并对从训练样本中选择出来的测试数据进行检测并计算误差

% 计算准确率

oobAccuracy = 1 - oobError(RF, 'mode', 'ensemble')

%计算参数对测试样本数据测试结果的准确率

[~,order] = sort(RF.OOBPermutedVarDeltaError); %矩阵分类

%将由训练样本数据所确定的随机森林参数按照由大到小的顺序进行排序

figure

barh(RF.OOBPermutedVarDeltaError(order))

%将各个参数通过二维直方图表示出来

set(gca,'YTickLabel',vars(order))

%按照匹配样本特征名称和其对于的参数值

```

计算得准确率为 79.7318%。

通过数据实验，得出以下关于随机森林与支持向量机在分类性能方面的几点结论：（1）使用随机森林无需预先对数据进行预处理，而若使用支持向量机则有必要进行数据预处理；（2）在二分类问题上，二者的泛化能力无显著差异；

（3）*k*-最近邻规则分类

```
Y_train = Train.Survived; %加载目标值
X_train = Train(:,3:end); %选择预测变量
vars = X_train.Properties.VariableNames; %提取变量名
X_train = table2array(X_train); %转换成数据矩阵
X_test = table2array(Test(:,2:end)); %转换成数据矩阵
c = cvpartition(Y_train, 'holdout', 0.30); %取出 30% 的数据用作交叉检验
%test_label=Y_train(test(c),:);
test_label=Y_train(631:891,:);

%predict_label = knnclassify(X_train(test(c,:), X_train(training(c,:),
Y_train(training(c,:), 6);

predict_label = knnclassify(X_train(631:891,:),
X_train(1:631,:), Y_train(1:631,:), 10);

accuracy = length(find(predict_label == test_label))/length(test_label)*100
```

计算得准确率为 69.7318%。

KNN 算法是根据计算两点之间的欧氏距离来判断分类的，将距离接近的点分为同一类，这原本没问题，但是在泰坦尼克问题则不适用。因为就性别而言，男女只相差 1，而票价却可以相差很大，从而导致票价之间的欧氏距离很大，进而导致票价的权重超过性别的权重，船舱等级同样如此。也就是说用 KN 算法分类时，加重了数值大的变量的权重，削弱了性别、船舱等级等变量的权重。但是实际上这是不合理的，因为性别和船舱等级是整个算法中最重要的变量，它们在很大程度上决定了乘客能够存活。所以 KN 算法效果差。

当然，可以给变量附加权重来改善这一情况，比如人为的加大性别、船舱等

级等变量的权重，削弱票价等的权重，这样可以有效改善辨识精度。

三、实验结论

对于泰坦尼克号游轮上游客的生存预测，通过把性别+姓的存活率作为一个特征，并增加相应类别的存活率这些特征，比如各种性别的存活率以及各种等级的存活率，把分类 ID 扔进模型之后有必要把所属 ID 的百分比加进去，通过 SVM 和随机森林法可以得到比较理想的预测结果。

四、小组分工：

程序：吴俊逸

参数调试：吴骏逸金宇尘

报告撰写：朱雯文叶朋飞赵帅