数据挖掘作业三

殷赫 生物医学工程 2220170871

数据集3: Titanic: Machine Learning from Disaster

**实验结果**

数据预处理

Embarked属性：缺失值采用众数填充，并转化为数值

df.Embarked[df.Embarked.isnull()] = df.Embarked.dropna().mode().values

df['Embarked'][df.Embarked=='C']=1

df['Embarked'][df.Embarked=='Q']=2

df['Embarked'][df.Embarked=='S']=3

df['Embarked']=df.Embarked.astype(int)

Age属性：采用随机数填充，并转换数据类型

randnum=np.random.randint(df['Age'].mean()- df['Age'].std(), df['Age'].mean() + df['Age'].std(), size=df['Age'].isnull().sum())

df['Age'][np.isnan(df['Age'])] = randnum

df['Age']=df.Age.astype(int)

sibsp、parch属性合并为Relatives属性

df['Relatives']=df['SibSp']+df['Parch']

Sex属性转换为数值

df['Sex'][df.Sex=='male']=1

df['Sex'][df.Sex=='female']=0

df['Sex']=df.Sex.astype(int)

分类模型分析

1）随机森林模型处理程序

random\_forest = RandomForestClassifier(n\_estimators=100)

random\_forest.fit(X\_train, Y\_train)

Y\_pred\_r = random\_forest.predict(X\_test)

2）线性回归模型处理程序

logreg = LogisticRegression()

logreg.fit(X\_train, Y\_train)

Y\_pred\_l = logreg.predict(X\_test)

3）分析结果

对性别预测结果分析

Gender\_grouped=X\_test.groupby(['Sex'])

G\_Survival\_Rate=Gender\_grouped.sum()/Gender\_grouped.count()

x=np.array([1,2])

width=0.3

plt.bar(x-width/2,G\_Survival\_Rate['R'],width,color='r')

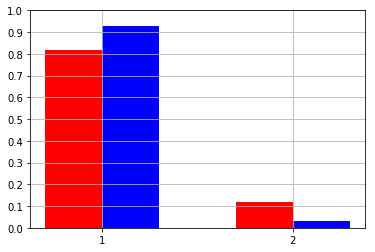
plt.bar(x+width/2,G\_Survival\_Rate['L'],width,color='b')

plt.xticks([1,2])

plt.yticks(np.arange(0.0, 1.1, 0.1))

plt.grid(True,linestyle='-',color='0.7')

plt.show()



纵轴为生存率，红色为随机森立预测结果，蓝色为线性模型预测结果，左侧为女性生存率预测，右侧为男性生存率预测。从结果可以看出女性生存率要远比男性生存率要高。

对舱类别预测结果分析

Pclass\_grouped=X\_test.groupby(['Pclass'])

P\_Survival\_Rate=Pclass\_grouped.sum()/Pclass\_grouped.count()

x=np.array([1,2,3])

width=0.3

plt.bar(x-width/2,P\_Survival\_Rate['R'],width,color='r')

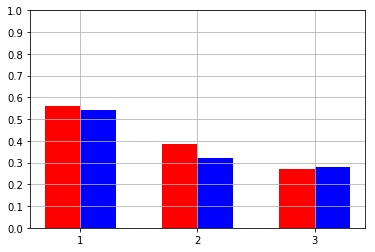
plt.bar(x+width/2,P\_Survival\_Rate['L'],width,color='b')

plt.xticks([1,2,3])

plt.yticks(np.arange(0.0, 1.1, 0.1))

plt.grid(True,linestyle='-',color='0.7')

plt.show()



纵轴为生存率，红色为随机森立预测结果，蓝色为线性模型预测结果，横轴从左到右,高等舱到低等舱。从结果可以看出高等舱要比低等舱存率要高。

聚类模型分析

1）Kmeans分析程序

Y\_cluster\_k = KMeans(n\_clusters=3, random\_state=10).fit(X\_cluster)

2）层次聚类分析程序

disMat = sch.distance.pdist(X\_cluster,'euclidean')

Z=sch.linkage(disMat,method='average')

P=sch.dendrogram(Z)

plt.savefig('plot.png')

cluster= sch.fcluster(Z, 1, 'inconsistent')

3）kmeans分析结果

对数据进行3分类结果如下

kmeans\_grouped=X\_cluster.groupby(['K'])

x=np.array([1,2,3])

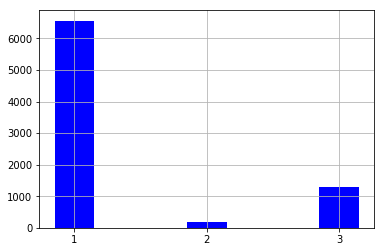
width=0.3

plt.bar(x,kmeans\_grouped.apply(np.size),width,color='b')

plt.xticks([1,2,3])

plt.grid(True,linestyle='-',color='0.7')

plt.show()



4）层次分析结果

