第五章

1.实践案例

train = pd.read\_csv(r"训练集数据地址.csv")

test = pd.read\_csv(r"测试集数据地址.csv")

full = pd.concat([train, test], axis=0, ignore\_index=True)

full['Cabin'] = full['Cabin'].fillna('U')

full['Embarked'].value\_counts()#查看总体乘客分别在S、C、Q上船的数量

full[(full['Survived']==1)&(full['Pclass']==1)&(full['Sex']=='female') ]['Embarked'].value\_counts()#加入条件，同类中S、C、Q乘客上船的数量

full['Embarked'] = full['Embarked'].fillna('S')#填充S（方法一）

full.loc[full['Embarked'].isnull(),'Embarked'] = 'S'#填充S（方法二）

temp=full[(full['Pclass']==3)&(full['Embarked']=='S')&(full['SibSp']==0) ]['Fare'].median()

full['Fare'] = full['Fare'].fillna(temp)

#同类中位数法

#以Pclassm,Parch,SibSp填充大部分数据：Pclass取[1,3]

#Pclass为[1,3]的值

for i in range(1, 4):

#取SibSp为[0,3]的值

for j in range(0, 4):

#Parch为[0,2]的值

for s in range(0, 3):

full1 = full.loc[(full['Pclass'] == i) & (full['SibSp'] == j) & (full['Parch'] == s) &(full['Age'].isnull()),'Age']

if len(full1) == 0:1

else:

large\_num = full.loc[(full['Pclass'] == i) & (full['SibSp'] == j) & (full['Parch'] == s) &(full['Age'].notnull()),'Age'].median()

full.loc[(full['Pclass'] == i) & (full['SibSp'] == j) & (full['Parch'] == s)

&(full['Age'].isnull()),'Age'] = large\_num

#对于Pclass=3，Parch=2,SibSp=8的个体，采用Pclass=3，Parch=2,SibSp=5的中位数进行填补

large\_age = full.loc[(full['Pclass'] == 3) & (full['SibSp'] == 5) & (full['Parch'] == 2) &(full['Age'].notnull()),'Age'].median()

print(large\_age)

full.loc[(full['Pclass'] == 3) & (full['SibSp'] == 8) & (full['Parch'] == 2) &(full['Age'].isnull()),'Age'] = large\_age

#观察到ID为1233和1256的乘客是8个孩子的父母，ID为1023的乘客是4个孩子的母亲，再通过观察同船其他乘客的数据，采用经验法对年龄缺失值进行填充

full.loc[1233,'Age'] = 45

full.loc[1256,'Age'] = 45

full.loc[1023,'Age'] = 40

Full['Title']=full['Name'].map(lambdax:x.split(',')[1].split('.')[0].strip())

print(full['Title'].value\_counts())

TitleDict={}

TitleDict['Mr'] = 'Mr'

TitleDict['Mlle'] = 'Miss'

TitleDict['Miss'] = 'Miss'

TitleDict['Master'] = 'Master'

TitleDict['Jonkheer'] = 'Master'

TitleDict['Ms'] = 'Mrs'

TitleDict['Mrs'] = 'Mrs'

TitleDict['Mme'] = 'Mrs'

TitleDict["Don"] = 'Royalty'

TitleDict["the Countess"] = 'Royalty'

TitleDict["Sir"] = 'Royalty'

TitleDict["Lady"] = 'Royalty'

TitleDict["Dona"] = 'Royalty'

TitleDict["Capt"] = 'Officer'

TitleDict["Col"] = 'Officer'

TitleDict["Major"] = 'Officer'

TitleDict["Dr"] = 'Officer'

TitleDict["Rev"] = 'Officer'

full['Title'] = full['Title'].map(TitleDict)

print(full['Title'].value\_counts())

sns.barplot(data=full,x='Title',y='Survived')

plt.show()

temp = full['Ticket'].value\_counts()

list\_shareticket = temp[temp>1].index

list\_shareticket = list\_shareticket.tolist()

full['isTicketShared'] = 0

full.loc[full['Ticket'].isin(list\_shareticket),'isTicketShared'] = 1

full.drop(['PassengerId','Name','Ticket','Cabin'],axis=1)

full=pd.concat([full,pd.get\_dummies(full['Sex'])],axis=1)

full=pd.concat([full,pd.get\_dummies(full['Title'])],axis=1)

full=pd.concat([full,pd.get\_dummies(full['Embarked'])],axis=1)

full = full.drop(['Sex','Title','Embarked'],axis=1)

print('Age的偏度：',stats.skew(full['Age']))

print('Fare的偏度：',stats.skew(full['Fare']))

full['Fare'] = np.log(full['Fare']+0.0001)

x = full.drop(['Survived'],axis=1)

y = full['Survived']

x = x.apply(lambda x: (x - np.mean(x)) / (np.std(x)))

full = pd.concat([y,x],axis=1)

clearned\_train = full[full['Survived'].notnull()]

clearned\_test = full[full['Survived'].isnull()]