**《Python高级应用》实验报告**

实验名称 ： 实验7 科学计算实验2

实验日期 ： 　2024/11/25

姓 名 ： 高心阳

学 号 ： 084623237

班 级 ： 计算机232

成 绩 ：

**人工智能与信息技术学院**

**南京中医药大学**

|  |
| --- |
| **实验目的：** |
| 1. 熟悉机器学习库的使用 2. 了解如何在实际中使用机器学习相关算法 |
| **实验内容和要求** |
| ‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬   1. Pandas库练习对泰坦尼克号的数据进行预处理；   （缺失值处理、归一化、文本数值化、特征合并、特征生成、主成分分析等）   1. 使用机器学习相关算法进行船员生死预测，并对预测结果进行衡量   （如使用逻辑回归、KNN算法、决策树、随机森林等算法，衡量的指标有准确率、召回率等）  3、（\*选做）去天池大数据搜索自己感兴趣的题目，了解刷题过程。 |
| **运行结果（写清题号）** |
| |  | | --- | | (1)代码 | | import pandas as pd  train\_file\_path = '../实验7科学计算实验2/train.csv' train\_data = pd.read\_csv(train\_file\_path) train\_data.columns = train\_data.columns.str.strip()  # 数据预处理 def preprocess\_data(data):  data = data.copy()  data.drop(columns=['Cabin'], inplace=True, errors='ignore')  data['Age'] = data['Age'].fillna(data['Age'].median())  data['Fare'] = data['Fare'].fillna(data['Fare'].median())   data['Embarked'] = data['Embarked'].fillna(data['Embarked'].mode()[0])   data['Sex'] = data['Sex'].map({'male': 1, 'female': 0})   data['Embarked'] = data['Embarked'].map({'S': 0, 'Q': -1, 'C': 1})   data['FamilySize'] = data['SibSp'] + data['Parch'] + 1   scaler = MinMaxScaler()   data[['Age', 'Fare']] = scaler.fit\_transform(data[['Age', 'Fare']])   return data  train\_data = preprocess\_data(train\_data) |  |  | | --- | | (2)代码 | | features = ['Pclass', 'Sex', 'Age', 'Fare', 'Embarked', 'FamilySize'] X = train\_data[features]  y = train\_data['Survived']  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42) X\_train = X\_train.fillna(X\_train.mean()) X\_test = X\_test.fillna(X\_test.mean())  models = {  "逻辑回归": LogisticRegression(),  "K近邻": KNeighborsClassifier(),  "决策树": DecisionTreeClassifier(),  "随机森林": RandomForestClassifier() } results = [] for model\_name, model in models.items():  model.fit(X\_train, y\_train)  y\_pred = model.predict(X\_test)  predictions\_df = X\_test.copy()  predictions\_df['真实值'] = y\_test.values  predictions\_df['预测值'] = y\_pred  predictions\_file = f"{model\_name}\_predictions.csv"  predictions\_df.to\_csv(predictions\_file, index=False)   acc = accuracy\_score(y\_test, y\_pred)  rec = recall\_score(y\_test, y\_pred)  precision = precision\_score(y\_test, y\_pred)  report = classification\_report(y\_test, y\_pred, output\_dict=True)  results.append({  "模型": model\_name,  "准确率": acc,  "召回率": rec,  "精确率": precision,  "分类报告": report  })  results\_df = pd.DataFrame(  [{"模型": r["模型"], "准确率": r["准确率"], "召回率": r["召回率"], "精确率": r["精确率"]} for r in results]) results\_df.to\_csv("model\_evaluation\_results.csv", index=False) |   结果如下：  模型 准确率 召回率 精确率  0 逻辑回归 0.809701 0.702703 0.812500  1 K近邻 0.802239 0.666667 0.822222  2 决策树 0.761194 0.702703 0.715596  3 随机森林 0.791045 0.738739 0.752294 |
| **实验的体会与建议** |
|  |