**沈思妤-10235501458-数据科学导论第七次作业**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.impute import SimpleImputer

from sklearn.metrics import f1\_score

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

df = pd.read\_csv('fraudulent.csv')

print(df.isnull().sum())

threshold = 0.2

df = df.loc[:, df.isnull().mean() < threshold]

imputer = SimpleImputer(strategy='most\_frequent')

df[df.columns] = imputer.fit\_transform(df)

X = df.drop(columns=['y'])

y = df['y']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=1)

# 逻辑回归

log\_model = LogisticRegression(max\_iter=1000, random\_state=1)

log\_model.fit(X\_train, y\_train)

log\_pred = log\_model.predict(X\_test)

log\_f1 = f1\_score(y\_test, log\_pred)

print(f'逻辑回归 F1 值: {log\_f1:.4f}')

# 决策树

tree\_model = DecisionTreeClassifier(random\_state=1)

tree\_model.fit(X\_train, y\_train)

tree\_pred = tree\_model.predict(X\_test)

tree\_f1 = f1\_score(y\_test, tree\_pred)

print(f'决策树 F1 值: {tree\_f1:.4f}')

# K-近邻

knn\_model = KNeighborsClassifier()

knn\_model.fit(X\_train, y\_train)

knn\_pred = knn\_model.predict(X\_test)

knn\_f1 = f1\_score(y\_test, knn\_pred)

print(f'K-近邻 F1 值: {knn\_f1:.4f}')