# 华为糖尿病预测实验工程报告

2022217586 邝黄硕

## 1. 引言

### 1.1 实验背景

糖尿病是一种严重的慢性疾病，影响了全球数百万人的生活质量。据美国疾病控制与预防中心（CDC）数据，美国目前约有1/7的成年人患有糖尿病，预计到2050年，这一比例将上升到1/3。通过数据挖掘和机器学习算法，可以对糖尿病进行早期预测，从而为预防和治疗提供有力支持。本实验利用UCI机器学习数据库中的印第安人糖尿病数据集，尝试建立一个糖尿病预测模型。

### 1.2 数据说明

数据集包含768个数据点，每个数据点有9个特征，分别是：怀孕次数、血糖、血压、皮脂厚度、胰岛素、BMI、糖尿病遗传函数、年龄和结果（0表示未患糖尿病，1表示患有糖尿病）。其中前8个特征均为数值型属性，最后一列为目标向量。

## 2. 实验方法

### 2.1 数据加载与初步探索

我们首先加载数据并进行初步探索，包括查看数据维度、数据样本、数据类型和目标变量分布。

import pandas as pd

df = pd.read\_csv(r'data/pima-indians-diabetes.data', sep=',')

print(df.shape)

print(df.head())

print(df.info())

print(df['Outcome'].value\_counts())

### 2.2 数据可视化

通过热力图和特征分布图对数据进行可视化，帮助理解特征之间的相关性及其分布情况。

import seaborn as sns

import matplotlib.pyplot as plt

plt.figure(figsize=(25, 15))

sns.heatmap(df.corr(method='spearman'), annot=True)

plt.show()

import math

def plot\_distribution(dataset, cols=5, width=20, height=15, hspace=0.2, wspace=0.5):

plt.style.use('seaborn-whitegrid')

fig = plt.figure(figsize=(width, height))

fig.subplots\_adjust(wspace=wspace, hspace=hspace)

rows = math.ceil(float(dataset.shape[1]) / cols)

for i, column in enumerate(dataset.columns):

ax = fig.add\_subplot(rows, cols, i + 1)

ax.set\_title(column)

if dataset.dtypes[column] == object:

g = sns.countplot(y=column, data=dataset)

substrings = [s.get\_text()[:18] for s in g.get\_yticklabels()]

g.set(yticklabels=substrings)

plt.xticks(rotation=25)

else:

g = sns.histplot(dataset[column], kde=True)

plt.xticks(rotation=25)

plot\_distribution(df, cols=3, width=20, height=20, hspace=0.45, wspace=0.5)

### 2.3 3D散点图

利用3D散点图对三个特征进行可视化，观察不同类别数据点的分布情况。

from mpl\_toolkits.mplot3d import Axes3D

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

for c, m, i, l in zip('rg', 'sx', np.unique(df['Outcome']), ['class\_1', 'class\_2']):

ax.scatter(df[df['Outcome'] == i].iloc[:, 0], df[df['Outcome'] == i].iloc[:, 4], df[df['Outcome'] == i].iloc[:, 6], c=c, marker=m, label=l)

ax.set\_xlabel(df.columns[1])

ax.set\_ylabel(df.columns[5])

ax.set\_zlabel(df.columns[7])

ax.set\_title("pima-indians-diabetes")

plt.legend()

plt.show()

### 2.4 特征降维

使用主成分分析（PCA）对特征进行降维，并绘制降维后的3D散点图。

from sklearn.decomposition import PCA

from sklearn import preprocessing

X = df.drop(columns=['Outcome']).values

y = df['Outcome'].values

X\_std = preprocessing.scale(X)

pca = PCA(n\_components=0.95)

X\_pca = pca.fit\_transform(X\_std)

print('the Top 95% variance\_ratio:', pca.explained\_variance\_ratio\_)

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

for c, m, i, l in zip('rg', 'sx', np.unique(y), ['class\_1', 'class\_2']):

ax.scatter(X\_pca[y == i, 0], X\_pca[y == i, 1], X\_pca[y == i, 2], c=c, marker=m, label=l)

ax.set\_xlabel('X\_pca1')

ax.set\_ylabel('X\_pca2')

ax.set\_zlabel('X\_pca3')

ax.set\_title("PCA")

plt.legend(loc='lower left')

plt.show()

### 2.5 数据集划分

将数据集划分为训练集和测试集，比例为9:1，并进行分层抽样，以保证训练集和测试集中各类别样本比例一致。

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(df.drop(columns=['Outcome']), df['Outcome'], test\_size=0.1, random\_state=42, stratify=df['Outcome'])

print('Distribution of y\_train {}'.format(Counter(y\_train)))

print('Distribution of y\_test {}'.format(Counter(y\_test)))

2.6 逻辑斯谛回归模型与交叉验证

训练逻辑斯谛回归模型，并使用交叉验证评估模型的预测性能。

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import cross\_val\_score

lr\_model = LogisticRegression()

cv\_score = cross\_val\_score(lr\_model, X\_train, y\_train, scoring='f1', cv=5)

print('cross validation score of raw model {}'.format(cv\_score))

### 2.7 超参数调优

使用网格搜索进行超参数调优，选择最佳的模型参数。

python

复制代码

from sklearn.model\_selection import GridSearchCV

c\_range = [0.001, 0.01, 0.1, 1.0]

solvers = ['liblinear', 'lbfgs', 'newton-cg', 'sag']

max\_iters = [80, 100, 150, 200, 300]

tuned\_parameters = dict(solver=solvers, C=c\_range, max\_iter=max\_iters)

grid = GridSearchCV(lr\_model, tuned\_parameters, cv=5, scoring='f1')

grid.fit(X\_train, y\_train)

print('best score for model {}'.format(grid.best\_score\_))

print('best parameters for model {}'.format(grid.best\_params\_))

print('best parameters for model {}'.format(grid.best\_estimator\_))

### 2.8 最终模型评估

使用最优参数训练最终模型，并评估其在训练集上的性能。

lr\_model\_final = LogisticRegression(C=1.0, max\_iter=80, solver='newton-cg')

lr\_model\_final.fit(X\_train, y\_train)

y\_train\_pred = lr\_model\_final.predict(X\_train)

print('final score of model version2: {}'.format(f1\_score(y\_train, y\_train\_pred)))

## 3. 结果与讨论

通过实验，我构建了一个糖尿病预测模型，并通过交叉验证和网格搜索对模型进行了优化。最终模型在训练集上的F1得分达到了较高水平，表明模型具有良好的预测性能。后续工作可以通过更多的特征工程和其他算法的尝试，进一步提升模型的性能。

## 4. 结论

本实验成功地利用印第安人糖尿病数据集建立了一个有效的糖尿病预测模型。通过数据可视化、特征降维、模型训练与调优，我们展示了如何使用机器学习方法进行二元分类问题的解决。未来的研究可以结合更多的临床数据和特征，进一步提高预测模型的准确性和泛化能力。