成人死亡率预测作业报告

洪嘉栋 学号2246005 工程师学院

摘要

本报告详细描述了完成《成人死亡率预测》作业的过程，包括数据预处理、模型选择、模型训练与评估。我们采用了中位数填充方法处理缺失值，并通过MLPRegressor作为回归模型，在训练集上达到了0.592的R2分数，在测试集上达到了57.02分，满足了最高分要求。

1. 数据预处理

1.1 缺失值处理

我们首先处理了数据中的缺失值。考虑到中位数对异常值不敏感，我们选择了中位数填充方法来填充缺失值。使用sklearn库中的SimpleImputer类，我们将缺失值替换为相应列的中位数。

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='median', missing_values=np.nan)
imputer = imputer.fit(data)
data = imputer.transform(data)
```

1.2 异常值处理

我使用了winsorize方法对异常值进行缩尾处理。

1.3 特征标准化

我们使用StandardScaler对数据进行标准化处理，以确保所有特征都在相同的尺度上。

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_norm = scaler.fit_transform(data)
```

2. 模型构建和评估

2.1 模型选择

我选择了MLPRegressor作为回归模型，因为它能够处理非线性关系，并且可以通过调整隐藏层和激活函数来提高模型的灵活性。

```
from sklearn.neural_network import MLPRegressor
regressor = MLPRegressor(hidden_layer_sizes=(200, 200, 200), max_iter=50,
activation='relu', alpha=0.01,
                         solver='adam', verbose=10, random_state=21, tol=1e-8)
```

2.2 模型训练

使用处理后的训练数据，我训练了MLPRegressor模型。在训练过程中，我调整了模型参数，如学习率、隐藏层数量和神经元数量，以优化模型性能。

```
def model_fit(train_data):
    train_y = train_data.iloc[:,-1].values
    train_data = train_data.drop(["Adult Mortality"], axis=1)
    train_data_norm, imputer, scaler = preprocess_data(train_data)

    train_x = train_data_norm.values
    regressor.fit(train_x, train_y)
    joblib.dump(regressor, 'model.pkl')
    joblib.dump(imputer, 'imputer.pkl')
    joblib.dump(scaler, 'scaler.pkl')

    return regressor
```

2.3 模型评估

我使用R2分数和均方误差（MSE）来评估模型性能。在训练集上，我的模型达到了0.592的R2分数和6339.163871247034的MSE。在测试集上，我的模型达到了57.02分。

```
from sklearn.metrics import r2_score, mean_squared_error
y_pred = regressor.predict(test_data)
r2 = r2_score(label, y_pred)
mse = mean_squared_error(label, y_pred)
print("MSE is {}".format(mse))
print("R2 score is {}".format(r2))
```

3. 模型调优和实验

3.1 模型调优

在训练集上，我尝试了更多层数的MLP网络，发现非常容易过拟合，训练集的精确度很高，但是测试效果很差，泛化性能不好。

3.2 其他模型尝试

在模型选择上，我也尝试了Lasso回归和Ridge回归，效果反而不如单纯线性回归。

```
from sklearn.linear_model import Lasso, Ridge
# Lasso回归
lasso = Lasso(alpha=0.05)
# Ridge回归
ridge = Ridge(alpha=0.05)
```

3.3 数据预处理尝试

我尝试了fancyimpute的SVT算法，并且尝试了预先将发达国家与发展中国家分开进行数据填充，结果都不如直接使用中值填充。

```
from fancyimpute import SoftImpute
imputer = SoftImpute(convergence_threshold=0.01, max_iters=100)
data[column_name] = imputer.fit_transform(data[column_name])
```

4. 结论

通过使用中位数填充方法处理缺失值，并采用MLPRegressor作为回归模型，我成功构建了一个能够准确预测成年人死亡率的模型。我的模型在测试数据上的表现达到了最高分要求，证明了方法的有效性。