作业 1：成人死亡率

姓名：钟洋　　学号：22224046
学院（系）专业：航空航天学院电子信息

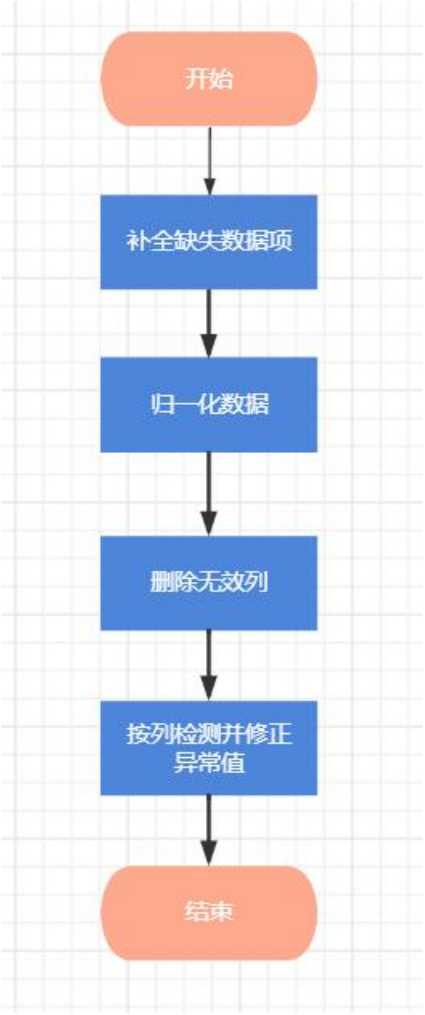
一、算法描述

1、算法整体的执行流程

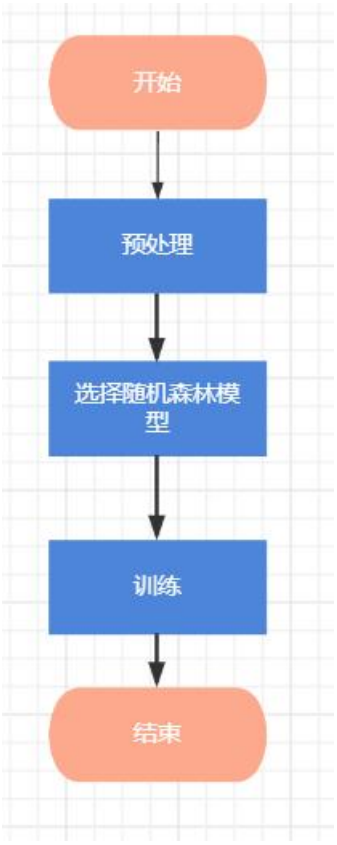
提交的 Python 文件中包含三个函数，分别是 preprocess、model_fit 和 predict。整个流程是首先利用 preprocess 函数预处理训练数据集，接着执行 model_fit 函数得到模型，然后提交到平台，平台调用 predict 函数利用前面训练的模型进行预测，最后评分。

2、各个函数的执行流程

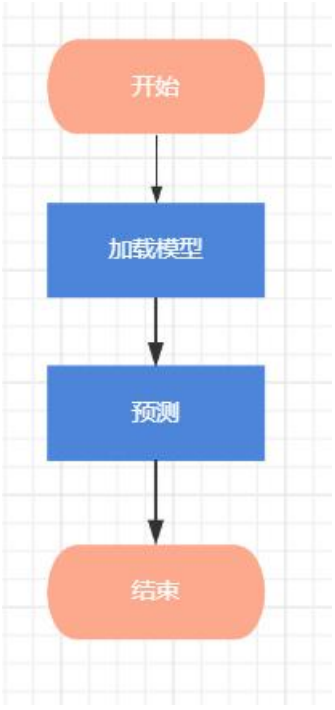
(1) preprocess 函数



(2) model_fit 函数



(3) predict 函数



3、源代码

(1) preprocess 函数

```
def preprocess_data(data, imputer=None, scaler=None):
    column_name = ['Year', 'Life expectancy ', 'infant deaths', 'Alcohol',
                   'percentage expenditure', 'Hepatitis B', 'Measles ', ' BMI ', 'under-five deaths ',
```

```
            'Polio', 'Total expenditure', 'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
            ' thinness  1-19 years', ' thinness 5-9 years', 'Income composition of resources',
            'Schooling']

    data = data.drop(["Country", "Status"], axis=1)

    if imputer==None:
        imputer = SimpleImputer(strategy='mean', missing_values=np.nan)
        imputer = imputer.fit(data[column_name])
    data[column_name] = imputer.transform(data[column_name])

    if scaler==None:
        scaler = MinMaxScaler()
        scaler = scaler.fit(data)
    data_norm = pd.DataFrame(scaler.transform(data), columns=data.columns)

    # 删除无效列
    data_norm = data_norm.drop(['Year', 'Measles ', 'Population'], axis=1)

    for column_idx in range(data_norm.shape[1]):
        column = np.empty(data_norm.shape[0], dtype=float)
        for i in  range(column.size):
            column[i] = data_norm.iloc[i, column_idx]

        # 异常值检测
        q75, q25 = np.percentile(column, [75 ,25])
        iqr = q75 - q25
        min_val = q25 - (iqr*1.5)
        max_val = q75 + (iqr*1.5)
        len_0 = len(np.where(column>max_val)[0])
        len_1 = len(np.where(column<min_val)[0])
        limit_0 = len_0 / len(column)
        limit_1 = len_1 / len(column)
        # 异常值处理
        new_column = winsorize(column,(limit_0, limit_1))
        data_norm.iloc[:, column_idx] = new_column

    joblib.dump(imputer, imputer_filename)
    joblib.dump(scaler, scaler_filename)

    return data_norm
```

(2) model_fit 函数

```
def model_fit(train_data):
```

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

# 线性回归
# regressor =linear_model.LinearRegression()

# SVM 回归
# regressor = svm.SVR()

# KNN 回归
# regressor = neighbors.KNeighborsRegressor()

# 随机森林
regressor = ensemble.RandomForestRegressor(n_estimators=25)

regressor.fit(train_x, train_y)
joblib.dump(regressor, model_filename)
return regressor
```

二、算法性能分析

1、线性回归

2、支持向量机

3、最近邻回归

4、随机森林回归

三、算法进一步研究展望

1、采用不同的数据填充方法，可以分别使用均值、中位数、众数和自定义常量来进行填充并比较算法结果，选择预测最准确的填充方法。

2、调整异常数据的检测与处理方法，比较算法结果，选择预测最准确的结果。