### 案例 2：基于 RF-GA 模型的城市二手房价格预测

* **问题背景**：二手房市场是房地产市场的重要组成部分，其价格受多种因素影响，如房屋面积、房龄、地理位置、周边配套设施（学校、医院、商圈等）、政策调控等。准确预测二手房价格，可为购房者、售房者和房地产政策制定者提供决策支持。
* **问题描述**：某房地产中介公司计划开发一套二手房价格评估系统，需要基于房屋的各项特征预测其成交价格。由于影响因素众多且关系复杂，传统单一模型难以精准预测，需采用优化后的集成模型。要求模型能自动优化参数，提高预测精度，帮助中介公司更合理地为房源定价。
* **数据情况**：提供该城市过去 3 年的二手房成交数据，每条数据包含房屋的基本属性（面积、户型、房龄、朝向等）、地理位置信息（经纬度、所属行政区等）、周边配套设施距离、成交时间、成交价格等。数据量约 5 万条，存在部分特征缺失和极端值。

### 案例 2：RF-GA 模型城市二手房价格预测代码

|  |
| --- |
| import pandas as pd  import numpy as np  from sklearn.preprocessing import LabelEncoder, MinMaxScaler  from sklearn.model\_selection import train\_test\_split  from sklearn.ensemble import RandomForestRegressor  from sklearn.metrics import mean\_absolute\_error  from geneticalgorithm import geneticalgorithm as ga  import joblib  # 数据加载与预处理  data = pd.read\_csv('second\_hand\_house.csv')  data = data.dropna()  # 类别特征编码  le = LabelEncoder()  for col in ['district', 'house\_type', 'orientation']:  data[col] = le.fit\_transform(data[col])  # 特征与目标变量  X = data.drop('price', axis=1)  y = data['price']  # 数据归一化  scaler = MinMaxScaler()  X\_scaled = scaler.fit\_transform(X)  # 划分训练集和测试集  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)  # 定义目标函数（RF模型MAE）  def objective\_function(params):  n\_estimators = int(params[0])  max\_depth = int(params[1])  min\_samples\_split = int(params[2])  min\_samples\_leaf = int(params[3])    rf = RandomForestRegressor(  n\_estimators=n\_estimators,  max\_depth=max\_depth,  min\_samples\_split=min\_samples\_split,  min\_samples\_leaf=min\_samples\_leaf,  random\_state=42  )  rf.fit(X\_train, y\_train)  y\_pred = rf.predict(X\_test)  return mean\_absolute\_error(y\_test, y\_pred)  # 定义参数范围  varbound = np.array([  [50, 500], # n\_estimators  [5, 50], # max\_depth  [2, 20], # min\_samples\_split  [1, 10] # min\_samples\_leaf  ])  # 遗传算法优化  algorithm\_param = {  'max\_num\_iteration': 50,  'population\_size': 20,  'mutation\_probability': 0.1,  'elit\_ratio': 0.05,  'crossover\_probability': 0.5,  'parents\_portion': 0.3,  'crossover\_type': 'uniform',  'max\_iteration\_without\_improv': None  }  model\_ga = ga(function=objective\_function,  dimension=4,  variable\_type='int',  variable\_boundaries=varbound,  algorithm\_parameters=algorithm\_param,  function\_timeout=500)  model\_ga.run()  best\_params = model\_ga.best\_variable  # 最优RF模型训练与预测  best\_rf = RandomForestRegressor(  n\_estimators=int(best\_params[0]),  max\_depth=int(best\_params[1]),  min\_samples\_split=int(best\_params[2]),  min\_samples\_leaf=int(best\_params[3]),  random\_state=42  )  best\_rf.fit(X\_train, y\_train)  y\_pred = best\_rf.predict(X\_test)  # 评估模型  print(f'最优参数: {best\_params}')  print(f'MAE: {mean\_absolute\_error(y\_test, y\_pred)}')  # 保存模型  joblib.dump(best\_rf, 'rf\_ga\_house\_price.pkl')  joblib.dump(scaler, 'scaler\_house.pkl') |