### 案例 7：基于 XGBoost - 贝叶斯优化模型的电商平台 “618” 大促销量预测

* **问题背景**：电商平台的大型促销活动（如 “618”“双十一”）期间，商品销量会出现爆发式增长，准确预测销量对平台库存管理、物流调度和营销策略制定至关重要。销量受商品价格、促销力度、用户评价、历史销量、竞争对手活动等多因素影响，呈现强非线性关系。
* **问题描述**：某大型电商平台需要对平台上 1000 款热销商品在即将到来的 “618” 大促期间（15 天）的日销量进行预测。要求模型能够自动优化超参数，充分利用高维特征数据，精准预测各商品的销量变化，避免库存积压或缺货。
* **数据情况**：提供各商品过去 3 年的日常销量及过往 “618”“双十一” 期间的销量数据，同时提供商品的属性信息（类别、价格、品牌、规格等）、促销方案（折扣力度、满减活动等）、用户评价数据（评分、好评率等）、平台流量数据等。数据量庞大，特征维度超过 50 维。

### 案例 7：XGBoost - 贝叶斯优化模型电商大促销量预测代码

|  |
| --- |
| import pandas as pd  import numpy as np  from sklearn.preprocessing import LabelEncoder, MinMaxScaler  from sklearn.model\_selection import train\_test\_split, cross\_val\_score  from xgboost import XGBRegressor  from skopt import BayesSearchCV  from skopt.space import Real, Integer  from sklearn.metrics import mean\_squared\_error  import joblib  # 数据加载与预处理  data = pd.read\_csv('ecommerce\_sales.csv')  data = data.dropna()  # 类别特征编码  le = LabelEncoder()  for col in ['product\_category', 'brand']:  data[col] = le.fit\_transform(data[col])  # 特征与目标变量  X = data.drop(['sales', 'product\_id'], axis=1)  y = data['sales']  # 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XGBoost参数空间  param\_space = {  'n\_estimators': Integer(50, 500),  'max\_depth': Integer(3, 10),  'learning\_rate': Real(0.01, 0.3, 'log-uniform'),  'subsample': Real(0.6, 1.0),  'colsample\_bytree': Real(0.6, 1.0),  'gamma': Real(0, 5)  }  # 贝叶斯优化  xgb = XGBRegressor(objective='reg:squarederror', random\_state=42)  bayes\_search = BayesSearchCV(  estimator=xgb,  search\_spaces=param\_space,  cv=5,  n\_iter=30,  scoring='neg\_mean\_squared\_error',  random\_state=42,  verbose=1  )  bayes\_search.fit(X\_train, y\_train)  best\_xgb = bayes\_search.best\_estimator\_  # 模型预测  y\_pred = best\_xgb.predict(X\_test)  # 评估模型  print(f'最优参数: {bayes\_search.best\_params\_}')  print(f'MSE: {mean\_squared\_error(y\_test, y\_pred)}')  # 保存模型  joblib.dump(best\_xgb, 'xgb\_bayes\_sales.pkl')  joblib.dump(scaler, 'scaler\_sales.pkl') |