Q4&Q5 分析报告

(一) 问题描述

性能评测 i. 在 a）的每日时间序列数据中，对每个商品按照安排时间从早到晚的顺序排列，分别选取该 商品 80%和 20%d 的时序数据作为训练和测试数据 ii. 对比仅使用 b.i 特征、仅使用 b.i+b.iv 特征、仅使用 b.i+b.iv+b.ii+b.v 特征、使用所有 b.i+b.iv+b.ii+b.v+b.iii+b.vi 特征等 4 类场景的性能对比，并加以讨论。 iii. 指标： root relative squared error (RSE)。

(二) 解决思路

在对回归问题的建模分析中，经常会遇到对回归问题的评估问题，常用的评估指标有 mean_squared_error，简称 **MSE**，即均方误差，计算公式为：

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

一般使用 **RMSE** 进行评估（这个回归分析模型中最常用的评估方法）：

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

而本项目中使用的 **RSE**（relative squared error）与 **RMSE** 的转换公式为：**RMSE / S**（**S** 为标准偏差），**S** 的计算公式为：

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})^2}$$

(三) RSE 计算结果

RSE表				
预测模型/特征工程	b.i	b.i+b.iv	b.i+b.ii+b.iv+b.v	b.i+b.ii+b.iii+b.iv+b.v+b.vi
SVM	0.729117943	0.751982519	0.751976322	0.751978753
DecisionTree	0.80793591	0.652352034	1.435126132	0.835142554
RandomForest	0.753130262	0.957913123	0.589479915	0.747336941
MLP	0.841594687	0.818134517	0.823332457	0.789756077

(四) 详细实现

- **辅助函数**

  `min_date()`:返回数据集中最早的日期

  `max_date()`:返回数据集中最晚的日期

  `get_date_list()`:获取返回从最早到最晚的所有日期数组

  `get_pluno_dict()`:获取返回以 `pluno` 为第一级 `key` 值，`date_str` 为第二级 `key` 值的字典

- **RSE 函数**

```
1.  def RSE():
2.      # SVM DecisionTree RandomForest MLP
3.      result_i = pd.read_csv('MLP/forecast_i.csv')
4.      result_i_iv = pd.read_csv('MLP/forecast_i_iv.csv')
5.      result_i_ii_iv_v = pd.read_csv('MLP/forecast_i_ii_iv_v.csv')
6.      result_all = pd.read_csv('MLP/forecast_all.csv')
7.      start = datetime.datetime.strptime('2016-02-01', '%Y-%m-%d')
8.      pluno_dict = get_pluno_dict()
9.      mean_y = 0
10.     predict_y = []
11.     true_y = []
12.     count = 0
13.     for index, row in result_all.iterrows():
14.         # 获取预测数据
15.         i = 3
```

```
        while i < len(row):
            predict_y.append(row[i])
            i += 1
        # 获取真实数据
        pluno = row['pluno']
        interval = row['time']
        date_time = start + datetime.timedelta(days=interval)
        end_time = start + datetime.timedelta(days=interval + 6)
        while date_time <= end_time:
            date_str = datetime.datetime.strftime(date_time, '%Y-%m-%d')
            true_y.append(pluno_dict[pluno][date_str])
            date_time += datetime.timedelta(days=1)
        # 计算销量平均值
        mean_y += row['qty']
        count += 1
    mean_y /= count
    # 计算 RMSE
    RMSE = np.sqrt(mean_squared_error(true_y, predict_y))
    # 求标准偏差
    Variance = 0
    num = 0
    for index, row in result_all.iterrows():
        Variance += np.square(row['qty'] - mean_y)
        num += 1
    Variance = np.sqrt(Variance / num)
    # 利用公式 RSE = RMSE / Variance 求解 RSE
    if Variance == 0:
        print("no way!")
        res = float('inf')
    else:
        res = RMSE / Variance
    print(res)
```

(五)　RSE 分析

首先可以看到使用不同模型、不同特征工程都会有不同的 RSE 值得变化，对于同一个模型来说，使用不同的特征工程得到得结果好坏情况不同。像 DecisionTree 模型在使用 b.i+b.iv 特征工程时结果最好为 0.652，而在使用 b.i+b.ii+b.iv+b.v 特征工程时预测结果反而十分糟糕，RSE 值为 1.43。而恰恰相反的是 RandomForest 模型，它在使用 b.i+b.ii+b.iv+b.v 特征工程时预测的结果最好，RSE 值为 0.589，而在使用 b.i+b.iv 特征工程时结果最差为 0.957。可见因

为不同模型的分析方法不同，导致模型对不同的特征量敏感性不同。

除此之外，通过 RandomForest、MLP 的 RSE 数据，可以看出随着特征量的增多，模型预测的准确性经历了一个下降后增加的过程。由此可见并不是特征量越多，得出得训练模型预测就越好，有时过多的特征值反而是误导，或者造成模型过度拟合，从而预测出的值脱离实际。

当然 RSE 指标只是评估回归预测模型好坏的一个指标，仅仅靠一个指标并不能严谨地评价模型的质量，而是要综合多角度的指标对模型进行一个综合的评估，没有完美的模型，我们应当根据我们的需求找到最适合使用的模型。