目录

摘要

配送服务是物流系统的重要组成部分,是面向客户的服务环节,配送的质量将直接影响着客户体验。精确地预测配送时间,提高货物配送的准时性,能够为客户节约时间,提升客户体验,有助于物流企业提高配送效率,降低配送成本,增强企业竞争力。因此,对于物流配送时间预测的研究具有重要意义[1]。

引言

在本次分析中，假设我作为一家食品配送公司的数据分析师，主要任务是找出送货员花费的时间与哪些因素相关，并尝试预测所用的时间。为了实现这一目标，我首先清除了数据集中的脏数据，并通过分析各个变量之间的影响因素，运用相关性系数来揭示它们之间的关系。随后，确定了预测模型所需的相关变量，并尝试了随机森林、XGBoost、LightGBM、VotingRegressor 模型融合等预测模型。评分结果表明，VotingRegressor 模型融合的预测精度更高，因此成为我们的首选模型。通过这次分析，我们可以为公司提供更准确的送货员花费时间预测，进而优化配送流程，提高效率。

正文

1 数据介绍

本次使用的食品配送数据集是来自 Kaggle 平台的开源数据集，该数据集有 20 个字段，共 45593 条记录。每条记录包含了唯一的标识。 目标是在给定相关信息的情况下，寻找发现各个字段与送货员花费的时间，之间的关系。和建立交付订单花费时间的预测模型。整个的分析和建模过程都是在 python3.7 环境中进行，所用的编辑器为 jupyter lab。

食品配送公司统计了配送订单的一些基本信息， 具体字段含义如表 1 所示：

字段	字段解释
objectID	表示条目的唯一标识
Delivery_person_ID	代表送货人员的唯一标识。
Delivery_person_Age	代表送货员的年龄。
Delivery_person_Ratings	表示给予送货员的平均评分。（1 到 5）
Restaurant_latitude	代表餐厅的纬度。
Restaurant_longitude	表示餐厅的经度。
Delivery_location_latitude	表示交货地点的纬度。
Delivery_location_longitude	表示交货地点的经度。
Order_Date	表示下订单的日期。
Time_Orderd	表示下订单的时间。
Time_Order_picked	表示从餐厅取餐的时间。
Weather	表示天气状况（有风、晴天、多云、暴风雨、雾、沙尘暴等）
Road_traffic_density	表示道路交通密度（Jam、High、Medium 和 Low）
Vehicle_condition	代表车辆的状况。（平滑、良好或一般）
Type_of_order	表示订单的类型（小吃、膳食、自助餐、饮料等）
Type_of_vehicle	代表正在使用的车辆类型（摩托车、自行车等）
multiple_deliveries	表示一次性尝试交付的订单数量
Festival	表示这一天是否节日
City	代表城市
Time_taken(min)	表示送货员花费的时间

表 1 字段说明

1.1 数据预处理

缺失值可视化，一般来说，未经处理的原始数据中通常会存在缺失值，这些缺失值会在画图可视化分析的时候影响程序的执行，同时也会影响预测模型，导致数据分析可靠性不强。因此，查看缺失值是数据预处理的第一步。导入 missingno 库，使用 msno.matrix(df)函数对缺失值进行可视化查看。如图 1-1 所示：

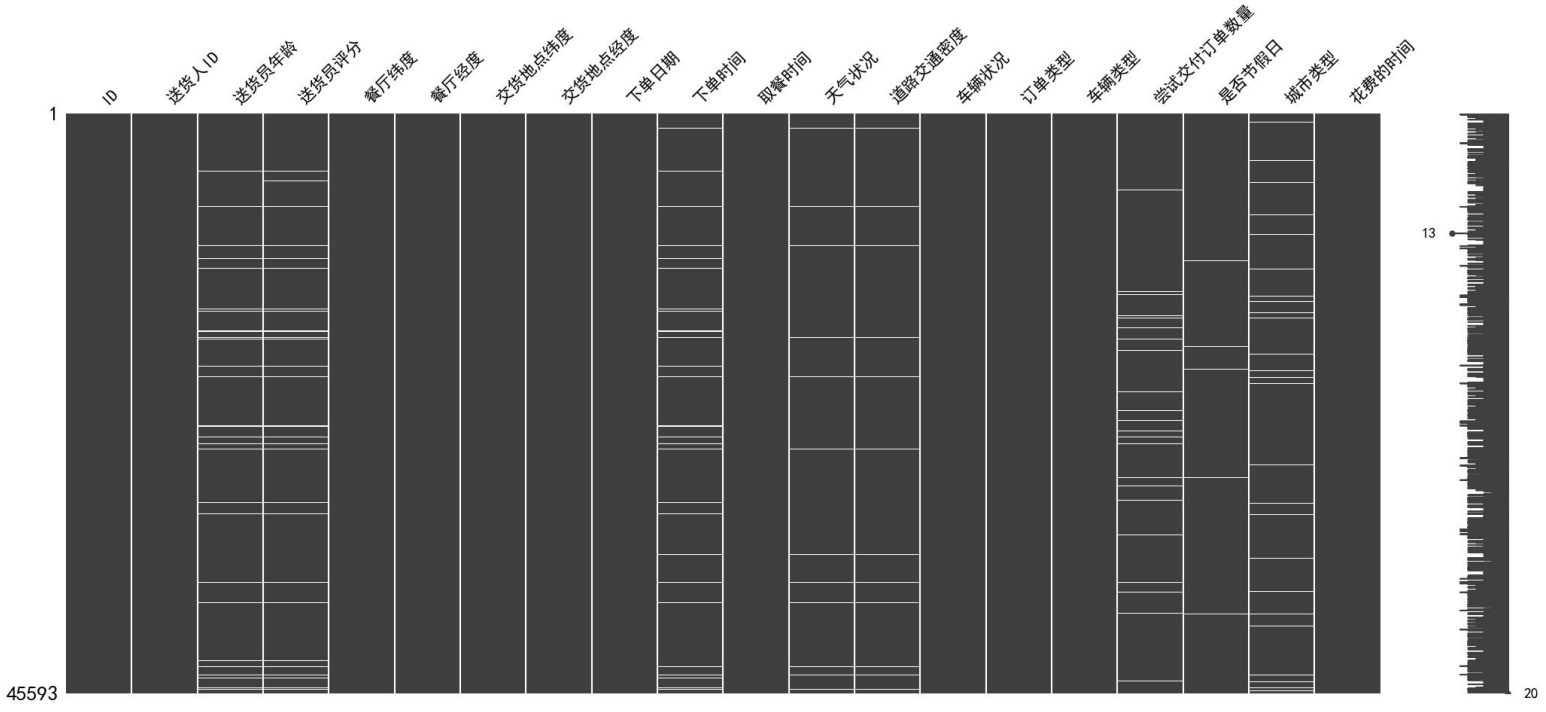

图 1-1 字段缺失值可视化图

观察图 1-1 字段缺失值可视化图可以发现，全部字段有效值覆盖率都在 80%以上还是比较高的。一部分字段有 10%以下的数据缺失，缺失数量较少，对 ['送货员年龄','送货员评分','尝试交付订单数量'] 连续型变量，数值类型用中位数进行缺失值填充。

对 ['是否节假日','城市类型','天气状况','道路交通密度'] 离散型变量，类别型用出现次数最多的值进行缺失值填充。将下单时间的空值，用取餐时间前十分钟时间填充处理。

并使用 df.drop_duplicates(inplace=True)方法对数据集每一行进行去重，对比前后数据行数发现无重复值。

1.2 异常值处理

异常值也通常被称为离群点，就是数据集中存在的明显不合理的点。在现实情况中不可能达到的值[2]。在进行数据预处理时，异常值是保留还是删除，需根据数据本身特点以及数据背景知识来判断，因为不是所有异常值都是无信息的，某些异常值也有可能包含有用信息。本文所使用的数据集中一些重要的连续型变量的描述性统计信息。df.describe(percentiles=[.1,.2,.3,.4,.5,.6,.7,.8,.9]).round().T 方法查看分布。 如图 1-2 所示：

列名	count	mean	std	min	10%	20%	30%	40%	50%	60%	70%	80%	90%	max
送货员年龄	45593.0	30.0	6.0	15.0	22.0	24.0	26.0	28.0	30.0	31.0	33.0	35.0	37.0	123.0
送货员评分	45593.0	5.0	12.0	1.0	4.0	4.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0	999.0
餐厅纬度	45593.0	17.0	7.0	0.0	11.0	12.0	13.0	17.0	19.0	20.0	22.0	23.0	27.0	31.0
餐厅经度	45593.0	71.0	21.0	0.0	73.0	73.0	74.0	76.0	76.0	77.0	78.0	78.0	80.0	88.0
交货地点纬度	45593.0	17.0	7.0	0.0	11.0	12.0	13.0	17.0	19.0	20.0	22.0	23.0	27.0	31.0
交货地点经度	45593.0	71.0	21.0	0.0	73.0	73.0	74.0	76.0	76.0	77.0	78.0	78.0	80.0	89.0
车辆状况	45593.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	2.0	2.0	2.0	3.0
尝试交付订单数量	45593.0	2.0	1.0	1.0	1.0	1.0	1.0	2.0	2.0	2.0	2.0	2.0	2.0	4.0
花费的时间	45593.0	26.0	9.0	10.0	15.0	18.0	20.0	23.0	26.0	28.0	30.0	34.0	40.0	54.0

图 1-2 连续型变量的描述性统计信息

在观察图表中的连续型变量描述性统计信息时，我们注意到在'送货员年龄'字段中存在异常值'123'，通常正常工作年龄不会超过 65 岁，因此我们将大于 65 岁的值用该字段的中位数进行填充。另外，在'送货员评分'字段中，出现了大于 5 分的异常值，由于评分应该在 1-5 分之间，因此我们将大于 5 分的分数转换为 5 分，以符合评分规则。这样的数据清理措施有助于确保数据的准确性和合理性。

1.3 查看连续型数据分布情况

使用 df.hist 方法绘制直方图来查看当前数据的分布情况是否正常。如图 1-3 所示：

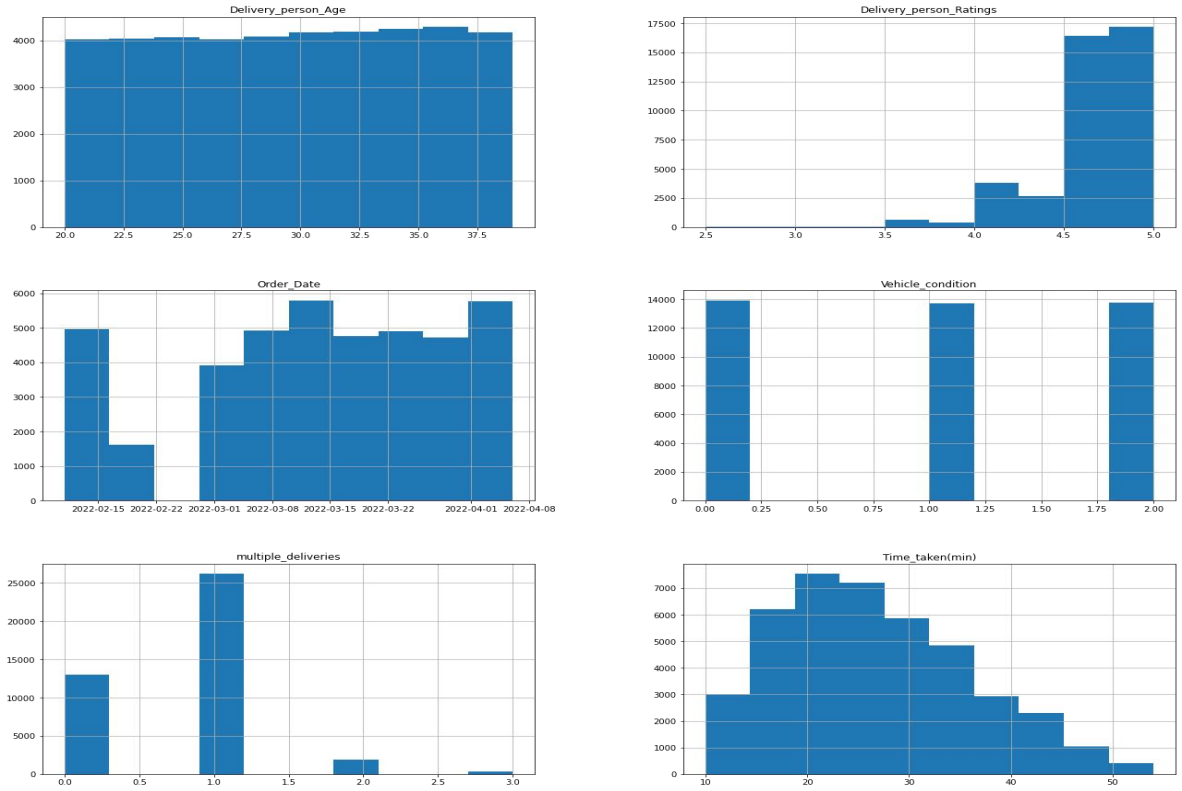

图 1-3 连续型变量分布情况

观察图 1-3 连续型变量分布情况，连续型变量分布正常，没有异常值。

2 变量分析

2.1 天气与花费的时间分析

小提琴图(Violin Plot)是用来展示多组数据的分布状态以及概率密度。这种图表结合了箱形图和密度图的特征，主要用来显示数据的分布形状。跟箱形图类似，但是在密度层面展示更好。下面利用小提琴图可视化分别查看天气与花费的时间，道路交通密度与花费的时间之间的情况如图 2-1，2-2 所示：

通过 Plotly Express 交互式可视化库使用 px.violin()函数来查看天气与花费的时间的分布情况如图 2-1 所示：

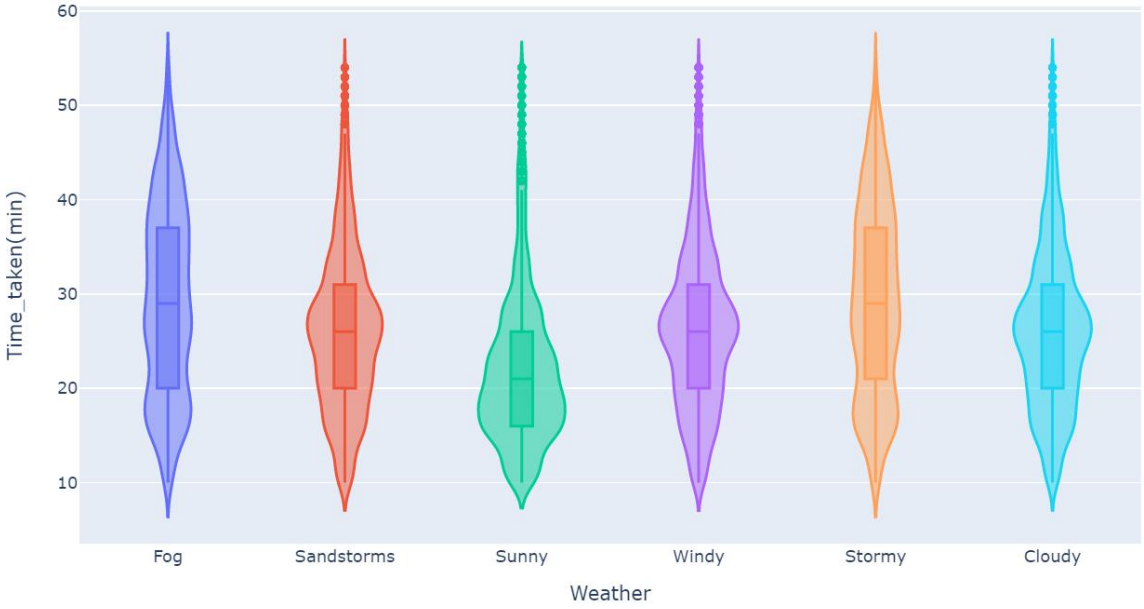

图 2-1 天气与花费的时间小提琴图

观察分析图 2-1 中的天气与花费的时间的小提琴图时，我们发现不同天气条件下，花费的时间存在一定的差异，尽管差异不是很显著。具体来说，雾天和暴风雨天的花费的时间中位数较高，且分布相对均匀。而晴朗天气的花费的时间中位数最低，主要集中在大约 17 分钟左右。其他天气条件下的花费的时间大多集中在 26 分钟左右。这符

合我们的一般经验，因为雾天和暴风雨天的视线较差，路况复杂，可能导致交付所用时间较长，而晴朗天气视线好，骑行更为方便。综合来看，从天气维度来看，晴朗天气下的花费的时间较短，而雾天和暴风雨天的花费的时间较长。

2.2 道路交通密度与花费的时间分析

道路交通密度与花费的时间的分布情况如图 2-2 所示：

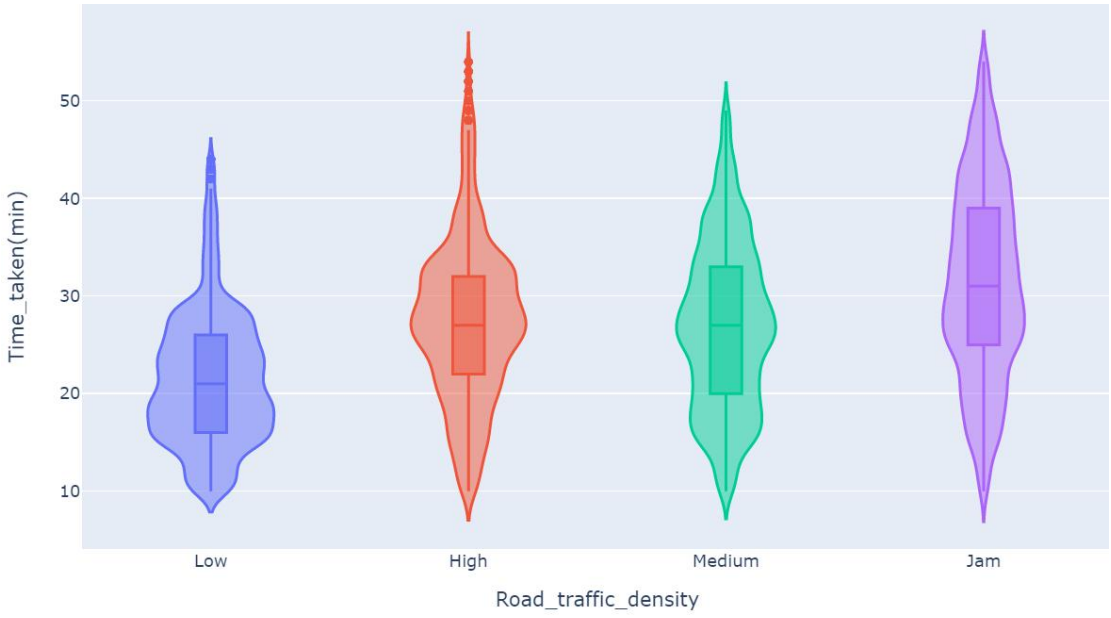

图 2-2 道路交通密度与花费的时间小提琴图

通过观察图 2-2 中的道路交通密度与花费的时间的小提琴图，我们发现不同路况下，花费的时间存在一定的差异。具体来说，低交通密度的路况下，花费的时间的中位数较低，大部分订单集中在 20 分钟以下。而拥挤的路况下，花费的时间的中位数最高，且分布相对均匀。其他路况的花费的时间大多集中在 27 分钟左右。

这符合我们的实际常识，因为在拥挤的路况下，送货员因道路拥堵可能会滞留更多时间，路况复杂等情况导致交付时间增加。而在低交

通密度的路况下，道路通畅便利，骑行速度较快，因此花费的时间较短。

综合来看，在道路交通密度维度下，低交通密度与拥挤交通密度的花费的时间存在明显差异，大致规律是道路交通密度越高，花费的时间越长。

2.3 车辆状况与花费的时间分析

核密度估计是估计随机变量的概率密度函数的非参数方法，即一种针对连续数据的密度估计方法，并且其根据数据本身的相互关系得到，无需对数据分布做假设[3]。核密度估计公式：

$$p(x) = \frac{1}{N}\sum_{k=1}^{n}\frac{1}{h}k(\frac{x - x_k}{h})$$

分析核密度图时主要是要观察其面积，对应的 y 轴取值是一个概率密度，只有与变量相乘才能得到该变量的概率取值。在做变量分析时，我们有时需要查看不同类别对应的花费的时间的概率分布密度，以此来找到有用的信息。下面利用核密度估计分别查看 车辆状况与花费的时间的情况，多次交货与花费的时间分析之间的情况， 如图 2-3，2-4，2-5 和 2-6 所示：

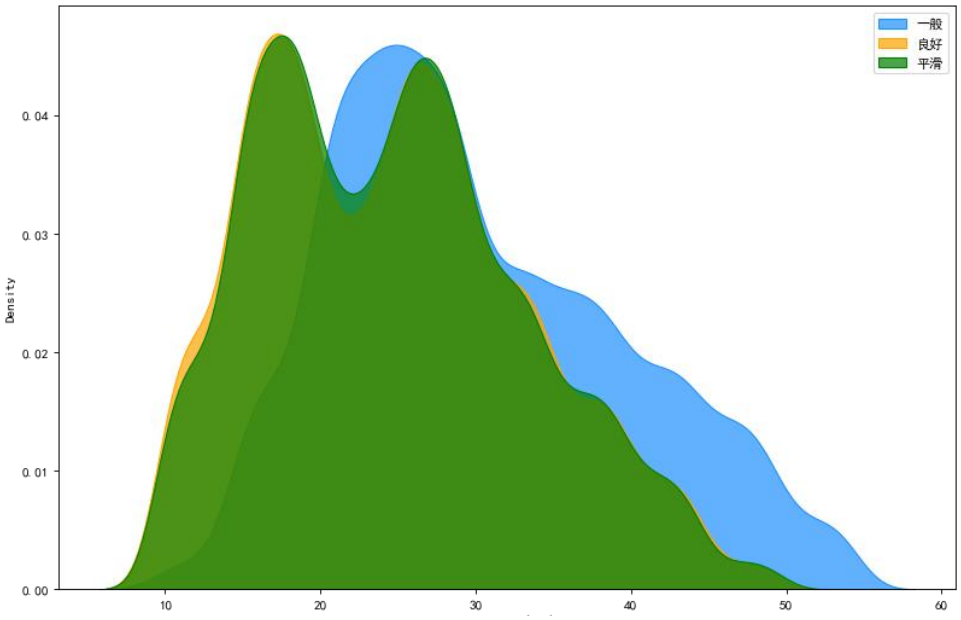

图 2-3 车辆状况与花费的时间核密度估计

观察图 2-3 中车辆状况与花费的时间的核密度估计，通过核密度面积分布图的分析，我们可以得出结论：车辆状况为一般的订单所用时间在一定概率下较车辆状况良好和平滑的订单更长。

这样的观察结果在实际情况中是合理的，因为车辆状况为一般的送货员可能面临更多的车辆故障等情况，导致花费的时间相对较长。而车辆状况良好和平滑的送货员可能更容易顺利完成送货任务，因此花费的时间相对较短。

2.4 多次交货与花费的时间分析

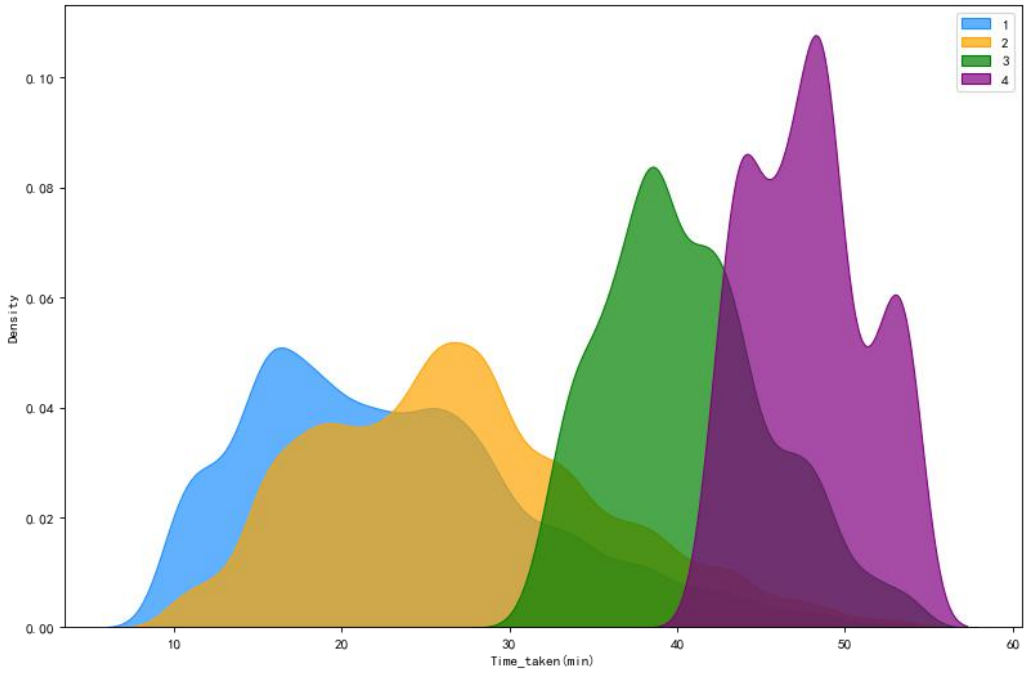

图 2-4 多次交货与花费的时间核密度估计

观察图 2-4 中多次交货与花费的时间的核密度估计，我们可以清楚地看到不同颜色所代表的订单数量分布错落有致。仔细观察可以发现，多次交货数量在核密度图上呈现从左到右逐渐递增的规律。

总体而言，随着一次性尝试交付的订单数量增加，花费的时间逐渐增加。这意味着在多次交货的情况下，送货员可能需要更多的时间来完成订单。这种现象的存在是合理的，因为多次交货可能意味着送货员需要在不同地点进行多次停车和交付，从而增加了交付订单所需的总时间。

2.5 节日与花费的时间分析

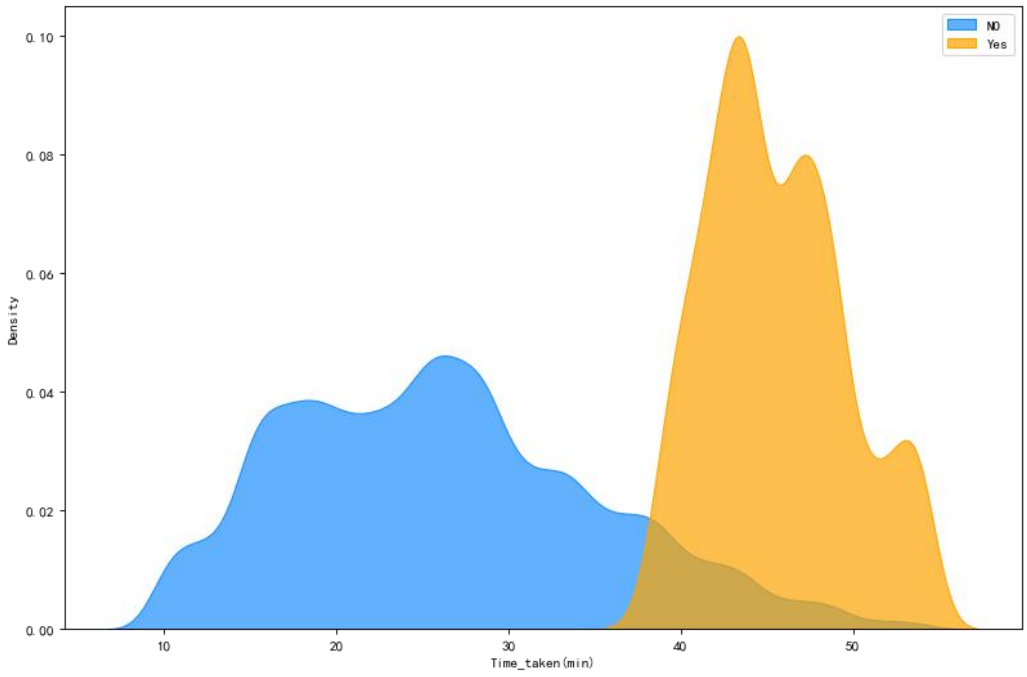

图 2-5 节日与花费的时间核密度估计

观察图 2-5 中节日与非节日期间花费的时间的核密度估计，我们可以得出以下结论：在非节日期间，花费的时间主要集中分布在 15-35 分钟之间；而在节日期间，花费的时间主要集中分布在 40-55 分钟之间。

总体而言，节日与非节日期间花费的时间存在较大的差异。节日期间花费的时间明显大于非节日期间花费的时间。这可能是因为在节假日期间，道路交通密度增加、停车难度加大以及人员流动性增加等因素导致送货员在交付订单时花费更多时间。

这一发现对于我们进行业务规划和资源调配具有重要意义，我们可以在节日期间提前安排更多送货员和优化路线规划，以确保订单的及时交付。

2.6 城市与花费的时间分析

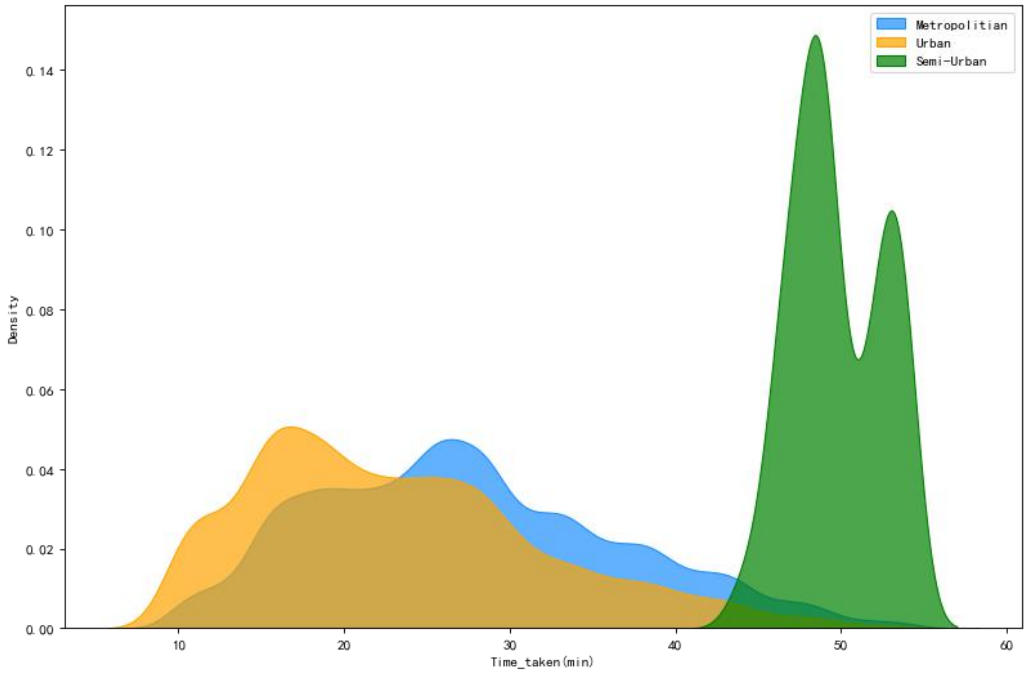

图 2-6 城市与花费的时间核密度估计

观察图 2-6 中城市类型与花费的时间的核密度估计，我们可以得出以下结论：在大都会和城市这两类城市中，花费的时间的分布没有太大差异。然而，在中小城市的情况下，花费的时间明显偏向右侧(更大值)。

为了找出造成这种差异的原因，我们对中小城市的各变量数据进行了进一步的分析。我们发现，中小城市的送货员 ID 去重后只有 156 人，仅占总样本送货员 ID 去重后的 5.6%。而中小城市里车辆状况为一般的占据了 85%以上。这可能是因为中小城市的送货员数量较少，配送人力有限，并且所配备的车辆状况较一般，这些因素导致了中小城市花费的时间的增加。

这一发现对于我们优化中小城市的配送服务和提升交付效率非常有价值。我们可以考虑增加送货员数量或改进车辆状况，以确保在中小城市地区的订单能够及时准确地完成交付。

2.7 分析交货地点的经纬度

图 2-7 交货地点热力地图

利用 Tableau 绘制交货地点经纬度的热力地图，我们可以直观地查看不同地区的订单数量密集度情况。大城市通常有更多的人口，自然而然地订单数量也会更多。通过热力地图的展示，我们可以直观地

了解订单的空间分布情况，有助于我们对不同地区的订单密集度有更深入的认识。

这样的分析可以为我们提供有价值的见解，例如在大城市的订单密集区域可能需要加强配送服务以满足需求，而在较偏远地区则可以优化配送策略以提高效率。通过利用数据分析工具，我们能够更好地优化配送服务，提升交付效率，为客户提供更好的服务体验。

3 特征工程与基准模型建立

特征工程指的是把原始数据转变为模型的训练数据的过程，目的就是获取更好的训练数据特征，使得机器学习模型逼近这个上限。特征工程能使得模型的性能得到提升，有时甚至在简单的模型上也能取得不错的效果。

3.1 离散型特征进行独热编码

在建立我们的机器学习模型之前，首先需要对数据集进行编码，因为我们的数据集包含了类别型的离散型特征。为了快速建立基准模型，我们可以采用一种简单粗暴的方法，即使用 OrdinalEncoder 进行编码。这样做的好处是节省了大量的代码和复杂的数据处理步骤。

这种方法相对简单，可以快速建立一个基准模型，以便在后续的特征工程和模型调优中进行对比和参考。这种简便的方式能够提供一个初始模型，使我们能够快速评估数据集和算法的潜在性能。

当然，在后续的工作中，我们还会进一步探索更精细的数据处理和特征工程方法，以提高模型的性能和泛化能力。

3.2 相关性分析

相关性分析作为一种常用的描述性分析方法。可以检查特征与特征之间的共线度，当共线度过大时，可能会引起模型的不稳定性。导致模型的鲁棒性差[4]。

将相关系数绘制成热力图，数值为两个变量之间的相关系数绘成颜色的深浅，从图中可以很方便的看出每个变量之间的相关性，以及相关性大小。

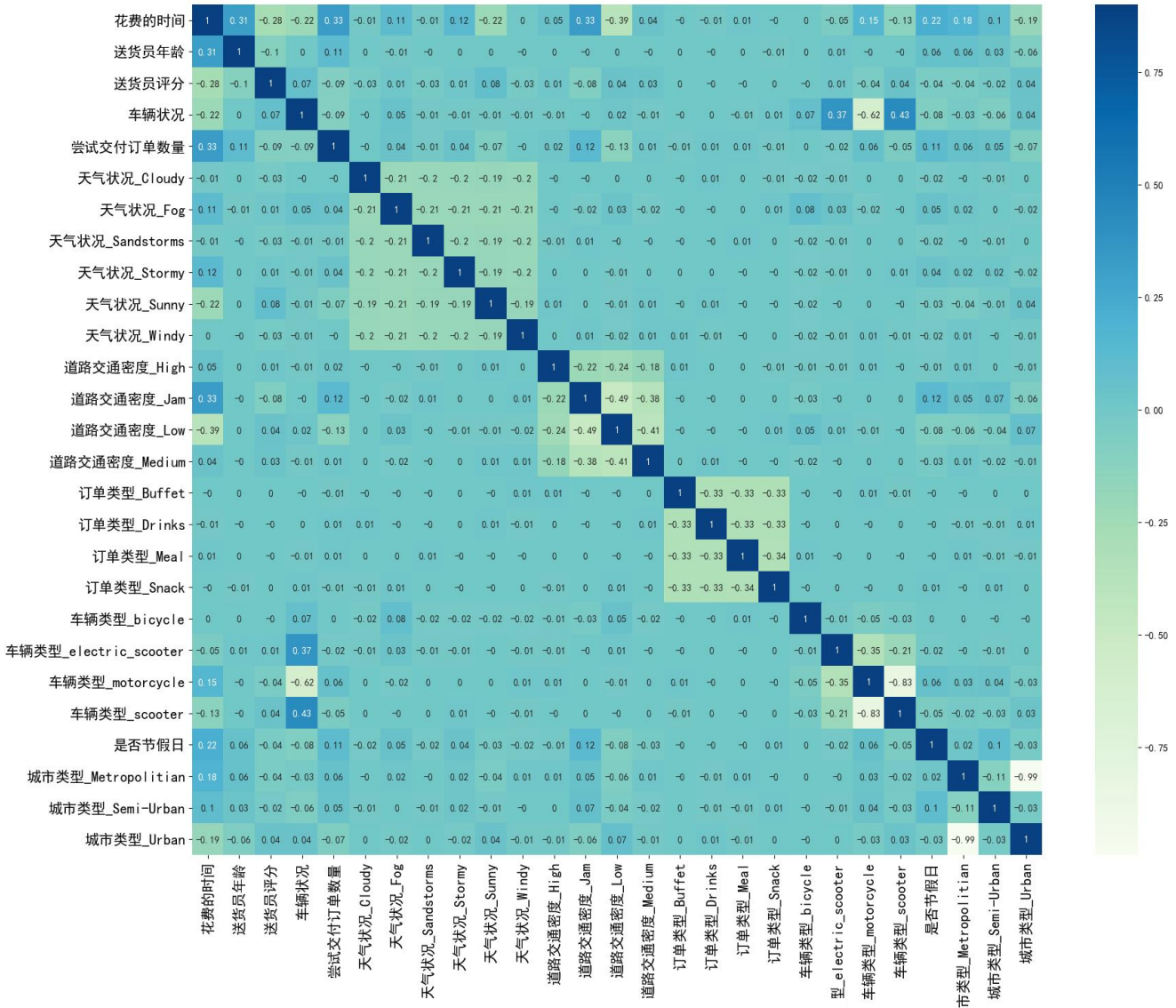

图 3-1 各变量之间的相关系数热力图

从严格的统计学意义讲，不同类型变量的相关性需要采用不同的分析方法，例如连续变量之间相关性可以使用皮尔逊相关系数进行计算，而连续变量和离散变量之间相关性则可以卡方检验进行分析，而离散变量之间则可以从信息增益角度入手进行分析。但是，如果我们只是想初步探查变量之间是否存在相关关系，则可以忽略变量连续/离散特性，统一使用相关系数进行计算，这也是 pandas 中的.corr 方法所采用的策略。所以我们使用 pandas 中的.corr 方法查看数据集中的相关系数。

观察分析上面图 3-1 各变量之间的相关系数热力图可看出

1.花费的时间与尝试交付订单数量呈现一定正相关，随着尝试交付订单数量的增加，交付时间可能会增加。

2.天气状况和道路交通密度与交付时间有关，雾、暴风雨天气以及拥挤的道路交通可能导致交付时间增加。

3.订单类型和车辆类型也与交付时间相关，不同订单类型和车辆类型可能会对交付时间产生影响。

4.城市类型也可能影响交付时间，大都会地区的交付时间可能较长。

3.3 建立基准模型

划分训练集和测试集。设置随机种子为 random_state=123。随机划分数据集后，建立随机森林回归模型，建立 XGBoost 回归模型，建立 LightGBM 回归模型。以上模型都以默认参数进行训练。

3.4 基准模型评分

使用 from sklearn import metrics 对各模型进行指标查看。三个模型的指标结果为表 2

	随机森林	xgboost	lightgbm
模型得分	0.8091	0.8172	0.8249
均方误差	16.686	15.978	15.300

表 2 基准模型指标

4 特征衍生

4.1 划分特征衍生数据集

划分 train、test 数据集方便后续的特征工程和数据预处理，从划分好数据集后开始，我们默认不知道 test 数据集中的预测目标特征‘花费的时间’，一切衍生特征信息都由 train 数据集产生并映射到 test 数据集。这样做是为了防止数据泄露，保留模型的泛化能力。

4.2 特征分箱

我们可以使用 KBinsDiscretizer 库对连续型特征进行离散化，将其转换为固定数量的区间或类别。为了实现这个目标，我们可以定义一个函数，用于对特定字段进行分箱。在这里，我们选取了字段"送货员年龄"和"送货员评分"作为例子，并分别进行聚类分箱，设定了箱子的数量。然后，我们将分箱结果和对应特征的平均数映射到训练集和测试集中，从而为后续的建模和分析提供离散化后的数据。这样的处理有助于在一些机器学习模型中更好地处理连续型数据。

4.2 聚合特征

在特征工程中，我们可以使用聚合特征来增强我们的特征矩阵。聚合特征是通过使用 groupby 函数对离散型变量进行聚合，从而获得新的特征。对于每个类别，我们可以得到一个具有业务意义的值。在实际应用中，增加大量的聚合特征可以显著提升树及集成模型的性能。因此，只要有业务意义，我们可以对任意特征进行聚合。在这个场景中，我们选择对目标特征"花费的时间"进行聚合，计算其平均值，以便为后续建模过程提供更加丰富的信息。

对以下字段['送货人 ID','天气状况','道路交通密度','车辆状况','订单类型','车辆类型','尝试交付订单数量','是否节假日','城市类型','小时','送货员年龄分箱','送货员评分分箱']使用 for 循环传入定义聚合花费的时间平均值特征矩阵函数中，train 数据集中生成的花费的时间平均值映射到 test 数据集中，并加 while 判断函数如果测试集中出现训练集中未出现的类别则打印提示。

4.3 简单类别字段的编码特征

定义简单编码函数，对以下字段进行简单编码

['天气状况', '道路交通密度', '订单类型', '车辆类型', '是否节假日', '城市类型']

为每个 train 数据集中的唯一类别分配一个数字代码(从 0 开始)，将映射存储在 dic 字典中。将存储在 dic 字典中的键值对再映射回 train，test 数据集里。

4.4 复杂类别字段的编码特征

在我们的数据集中，"送货人 ID"是一种非常特殊的离散型特征。在一般的数据分析中，带有唯一性的特征（如 ID）通常会被直接删除或忽略，因为它们在建模过程中往往没有太大的意义。然而，在我们当前的预测场景下，订单的送货速度可能因每个送货人的个性而异，因此我们需要保留并探索这个特征。

这个特征具有以下两个特点：

1.可能存在某些送货人 ID 只对应一个订单样本的情况。

2.在一个收货的城市类型下，可能会有几十上百个订单，因此订单中的花费时间均值可以被认为是经过大量实验后的可靠结果。

(df2.groupby('送货人 ID')['ID'].count()).describe()使用此方法检查送货人 ID 最少的配送订单，发现最小值为 5，在这里我们默认订单足够多。

在与其他离散型特征相比较时，送货人 ID 特征中的类别数量明显比较多。在分割训练集与测试集时，通常我们会假设离散字段中的所有类别都会同时出现在训练集和测试集中，以便使用相同的字典对字段进行编码。

然而，当一个离散字段中的类别数量非常多时，我们无法保证所有的类别会同时出现在训练集和测试集中。这就可能导致在测试集中出现了训练集中从未见过的分类或数据，这样的情况会使得我们不能再使用相同的字典对数据进行编码，否则会产生空值。

为解决这个问题，我们需要找到一种适当的方法来处理测试集中出现的新类别。通常的做法是对这些新类别进行特殊标记或进行合理的处理，以确保模型在测试集上的稳健性和准确性。这样能够保证我们在训练和预测过程中都能充分利用数据，避免出现缺失或错误的编码结果。

在此我们定义专门处理送货人 ID 编码的函数，将送货人 ID 进行简单编码后，将测试集中缺失值的行筛选出来，并获取该列中缺失值所对应的唯一类别。将该类别加入到字典 dic 中，并分配一个新的编号，使用字典 dic 将测试集中的缺失值所对应的类别进行映射替换。将替换后的结果赋值回测试集中的原始列。

完成后检查测试集，无空值。至此完成特征工程任务。

5 模型调优与模型融合

5.1 模型调优

划分出训练集测试集及其标签后，代入模型中进行模型调优。

画出随机森林 max_depth（最大深度）的学习曲线，观察出取值在 12 时模型没有严重过拟合，MSE 达到比较低的数值。所以我们 max_depth 参数选取 12

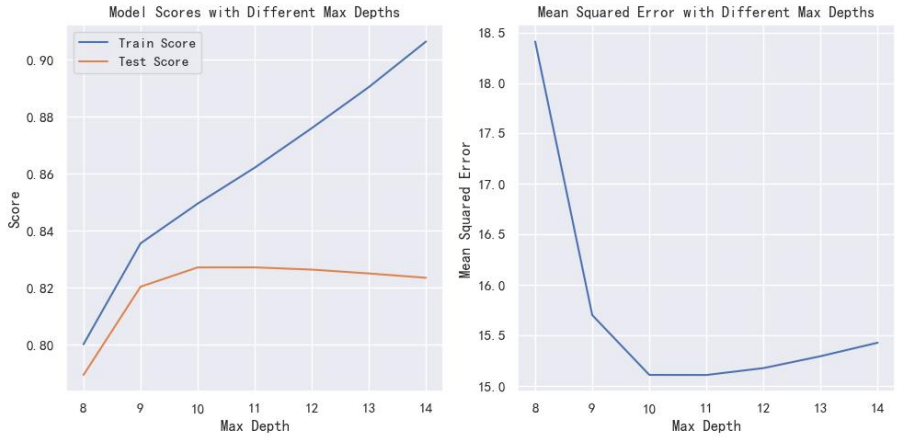

图 5-1 随机森林 max_depth 学习曲线

对 lightgbm 模型里的 max_depth 不同取值绘制学习曲线，观察出取值在 16 时模型评分达到高值，MSE 达到比较低的数值。所以我们 max_depth 参数选取 16。

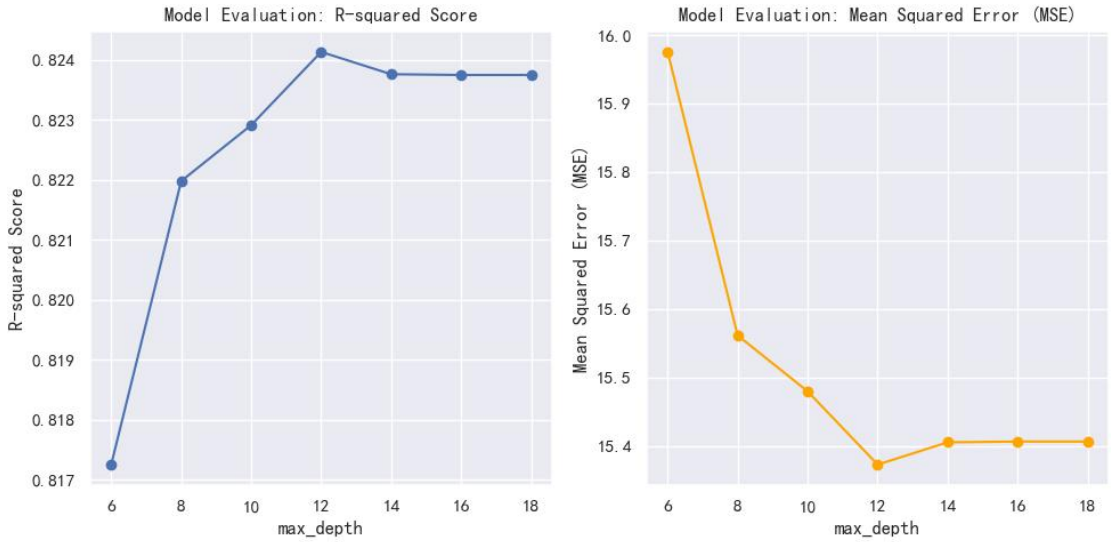

图 5-2 lightgbm max_depth 学习曲线

对 lightgbm 模型里的 min_child_samples 不同取值绘制学习曲线，观察出取值在 40 时模型评分达到高值，MSE 达到比较低的数值。所以我们 min_child_samples 参数选取 40。

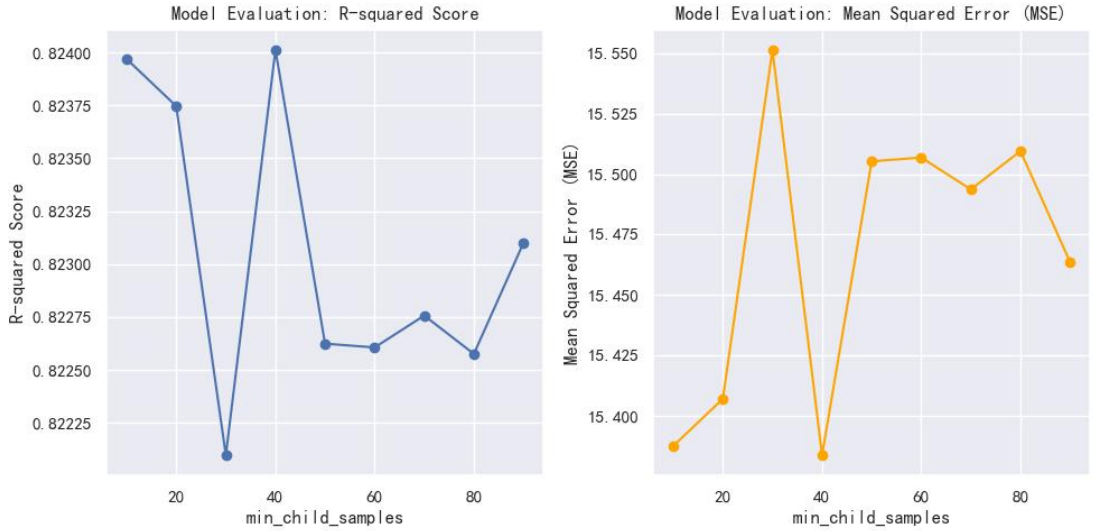

图 5-3 lightgbm min_child_samples 学习曲线

5.2 模型融合

我们使用了三个不同的基分类器（RandomForestRegressor、XGBRegressor 和 LGBMRegressor）来构建一个投票回归器（VotingRegressor）。这个投票回归器采用了投票策略，即多个基分类器对最终预测结果进行投票，以得到更加稳健和准确的预测。我们首先实例化了三个基分类器，并设置了它们的不同参数。然后，我们将这些基分类器组合在一起，形成了投票回归器（voting_reg）。计算预测结果与实际值之间的 R2 评分和均方误差（Mean Squared Error）。R2 评分用于衡量模型对测试集数据的拟合程度，越接近 1 表示模型拟合效果越好。均方误差则用于衡量预测值与实际值之间的偏差，值越小表示模型预测的准确性越高。

通过使用投票回归器结合多个基分类器的结果，我们期望能够获得更加稳健和准确的预测模型，从而在实际业务场景中取得更好的预测效果。这样的集成方法常常能够在不同基分类器之间相互补充，提升整体性能，对于解决复杂的预测问题有很好的应用价值。

	随机森林	xgboost	lightgbm	VotingRegressor
模型得分	0.8268	0.8121	0.8240	0.8271
均方误差	15.136	16.418	15.383	15.105

表 3 各模型指标

在本次数据分析任务中，我们使用了三种不同的回归模型：随机森林(Random Forest)、XGBoost 和 LightGBM，以及使用这三个模型集成的投票回归器（VotingRegressor）。我们看到各个单独模型的评估结果如 表 3 相对与基准模型，效果有微小的提升。

根据评估结果来看，VotingRegressor 模型在预测任务上表现稍微优于单独的三个模型。然而，它们之间的性能差异并不是很大。随机森林和 LightGBM 模型在得分和均方误差上表现相似，而 XGBoost 模型稍微略逊一筹。

综合考虑，我们可以得出结论：在本次预测任务中，集成模型 VotingRegressor略优于单独的随机森林、XGBoost 和 LightGBM 模型。但是，所有模型的性能都在可接受范围内，因此我们可以根据实际需求和应用场景来选择合适的模型。

6 结论建议与展望

6.1 结论

变量分析的结论：

天气方面：雾(Fog)和暴风雨(Stormy)天气的花费时间普遍较长，而晴朗(Sunny)天气的花费时间最短。

路况密度方面：低交通密度(Low)的路况下花费时间普遍较短，而拥挤(Jam)路况下花费时间普遍较长。

车辆状况方面：一般车辆状况的花费时间略多于其他车辆状况。

尝试交付订单数量：随着一次性尝试交付的订单数量的增加，花费时间逐渐增加。

节假日：节日的花费时间明显高于非节假日。

城市类型：中小城市的花费时间明显高于其他城市类型，主要城市的订单数量多于偏远城市的订单数量。

相关系数分析结论：

花费的时间与尝试交付订单数量呈现一定的正相关关系，随着尝试交付订单数量的增加，交付时间可能会增加。

天气状况和道路交通密度与交付时间有关，雾、暴风雨天气以及拥挤的道路交通可能导致交付时间增加。

订单类型和车辆类型也与交付时间相关，不同订单类型和车辆类型可能会对交付时间产生影响。

城市类型也可能影响交付时间，大都会地区的交付时间可能较长。

花费的时间预测模型构建的总结：我们构建了基于VotingRegressor 模型的花费的时间预测模型，该模型表现出良好的精度，平均误差在 4 分钟以内。预测结果的准确性足以满足大部分实际配送需求，如果将预计送达时间的预测精度扩大到±8 分钟范围内。将能够覆盖绝大部分订单真实的所用时间。

该模型对送货员的订单配送时间预测提供了有力支持，同时也方便了用户对下单后的订单收货时间有所了解。提前预测花费的时间有助于提高配送效率，节省时间成本。

6.2 建议

选择合适的配送时间。配送时间的选择对于提高车辆利用率有非常重要的作用。对于规定了装卸点的配送线路中，在一定的配送时间

内虽然多跑了距离，但节省了配送时间， 变相的提高了车辆利用率，节约了运输成本[5]。

1.在中小城市增加送货员，以提高送货效率。

2.优化车辆状况，确保车辆在良好及以上状态，保证交付过程的顺利进行。

3.配送时优先选择道路密度低的道路行驶，避免拥挤路段，减少交通延误。

4.预测时间过长时，减少交付订单数量，避免超负荷配送造成时间浪费。

5.在节日前提前准备热销货物，做好充分准备，提高配送效率。

6.建立送货历史记录，通过数据分析了解每个周期不同地点的产品销量情况，合理安排配货地点，进一步提高配送效率。

6.3 展望

在本次分析中，我们注意到数据的时间跨度较小，只有 2022 年的 2、3、4 月份数据，这可能限制了我们对整体趋势的把握。未来，我们可以争取获得更多时间跨度更广的数据，以提高分析的全面性和准确性。

另外，我们的预测模型对花费的时间预测精度尚有提升空间，为进一步提高预测精度，我们可以进行参数调优，尝试不同的模型算法，并进行特征工程优化，以寻找更合适的模型配置和特征组合。

通过不断优化和改进，我们相信可以进一步提高模型的预测准确性，从而为送货员提供更精准的配送时间预测，优化配送流程，提高客户满意度。

在未来，我计划深入学习深度学习技术，以处理更大规模、更复杂的数据。同时，我对学习不同的机器学习算法模型也充满兴趣，我的目标是在接下来的日子里，持续学习数据分析领域的知识，并将其应用到日常生活和工作中。通过不断学习和进步，我希望能够不断提升自己在数据分析领域的能力和水平。

参考文献

[1] 常正阳. 基于 SVR--Kalman 滤波的物流配送时间预测[D]. 西安科技大学.
[2] 申传华. 数据挖掘过程中的数据清洗研究[J]. 通讯世界,2016(24).
[3] 徐思琪. 基于核密度估计的半监督特征选择.
[4] 刘振江.影响巷道出口温湿度的单因素分析及正交实验研究[D].青岛理工大学,2019.
[5] 王勇,池洁.物流配送路线及配送时间的优化分析[J].重庆交通大学学报(自然科学版).2008,(27)卷第 4 期:648