接下来，我们将对出进站流量数据进行合并：

```
data_inNums.rename(columns={'time_10_minutes':'startTime'}, inplace=True)
data_outNums.rename( columns={'time_10_minutes':'startTime'}, inplace=True)
df_data = df_data.merge(data_inNums, on=['stationID', 'startTime'], how='left')
df_data = df_data.merge(data_outNums, on=['stationID', 'startTime'], how='left')
df_data['inNums']  = df_data['inNums'].fillna(0)
df_data['outNums'] = df_data['outNums'].fillna(0)
```

- 特征提取

在 baseline 部分，仅提取些基础特征即可。对于时间序列预测问题，主要提取简单的时间特征和历史平移特征。下面是提取时间相关特征的具体代码：

```
# 时间相关特征
df_data['time'] = pd.to_datetime(df_data['startTime'])
df_data['days'] = df_data['time'].dt.day
df_data['hours_in_day'] = df_data['time'].dt.hour
df_data['day_of_week'] = df_data['time'].dt.dayofweek
df_data['ten_minutes_in_day'] = df_data['hours_in_day'] * 6 + df_data['time'].dt.minute // 10
del df_data['time']
```

用于描述当前时间在所处周期内位置信息的特征是非常具有套路性的，其作用也非常大。比如星期特征（day_of_week）有助于发现相同星期数具有的相似性，类似周期性和相关性描述。下面是提取历史平移特征的具体代码：

```
# 历史平移特征
df_data['bf_inNums'] = 0
df_data['bf_outNums'] = 0
for i, d in enumerate(days):
    If d == 1:
    continue
    df_data.loc[df_data.day==d, bf_inNums] = df_data.loc[df_data.day==days[i-1], inNums]
    df_data.loc[df_data.day==d, bf_outNums] = df_data.loc[df_data.day==days[i-1], outNums]
```

- 模型训练

为了快速生成一个可靠稳定的结果，我们选择使用 LightGBM 模型，线下验证方式采用时序验证策略，用 1 月 28 日的刷卡数据作为验证集。模型训练的代码如下：

```
# 训练集和验证集准备
cols = [f for f in df_data.columns if f not in ['startTime','endTime','inNums','outNums']]
df_train = df_data[df_data.day<28]
df_valid = df_data[df_data.day==28]

X_train = df_train[cols].values
X_valid = df_valid[cols].values

y_train_inNums = df_train['inNums'].values
y_valid_inNums = df_valid['inNums'].values
```