数据挖掘互评作业 3：分类与预测

姓名：曹健　　学号：3120190978

一．数据

1. 数据集选择：**Hotel booking demand**
2. 数据集描述：
   该数据集包含城市酒店和度假酒店的预订信息，包括预订时间、停留时间，成人/儿童/婴儿人数以及可用停车位数量等信息

二．问题

基于这个数据集，进行以下问题的探索：

- 基本情况：城市酒店和假日酒店预订需求和入住率比较；
- 用户行为：提前预订时间、入住时长、预订间隔、餐食预订情况；
- 一年中最佳预订酒店时间；
- 利用 Logistic 预测酒店预订。

三、数据挖掘

1. 缺失值处理

```
# 缺失值处理
print("缺失值处理前：")
print(df.isnull().any()) # 每列是否有缺失值
print(df.isnull().sum()) # 每列的缺失值总行数
nan_replace = {"children": 0,"country": "Unknown", "agent": 0, "company": 0}
df_cln = df.fillna(nan_replace)
df_cln["meal"].replace("Undefined", "SC", inplace=True)
zero_guests = list(df_cln.loc[df_cln["adults"]
                 + df_cln["children"]
                 + df_cln["babies"]==0].index)
df_cln.drop(df_cln.index[zero_guests], inplace=True)
df = df_cln
print("缺失值处理后：")
print("df.shape:\n", df.shape)  #  (119390, 32)
print(df.isnull().any()) # 每列是否有缺失值
print(df.isnull().sum()) # 每列的缺失值总行数
```

```
缺失值处理前：
hotel                           False
is_canceled                     False
lead_time                       False
arrival_date_year               False
arrival_date_month              False
arrival_date_week_number        False
arrival_date_day_of_month       False
stays_in_weekend_nights         False
stays_in_week_nights            False
adults                          False
children                         True
babies                          False
meal                            False
country                          True
market_segment                  False
distribution_channel            False
is_repeated_guest               False
previous_cancellations          False
previous_bookings_not_canceled  False
reserved_room_type              False
assigned_room_type              False
booking_changes                 False
deposit_type                    False
agent                            True
company                          True
days_in_waiting_list            False
customer_type                   False
adr                             False
required_car_parking_spaces     False
total_of_special_requests       False
reservation_status              False
reservation_status_date         False
dtype: bool
```

```
缺失值处理后：
df.shape:
 (119210, 32)
hotel                           False
is_canceled                     False
lead_time                       False
arrival_date_year               False
arrival_date_month              False
arrival_date_week_number        False
arrival_date_day_of_month       False
stays_in_weekend_nights         False
stays_in_week_nights            False
adults                          False
children                        False
babies                          False
meal                            False
country                         False
market_segment                  False
distribution_channel            False
is_repeated_guest               False
previous_cancellations          False
previous_bookings_not_canceled  False
reserved_room_type              False
assigned_room_type              False
booking_changes                 False
deposit_type                    False
agent                           False
company                         False
days_in_waiting_list            False
customer_type                   False
adr                             False
required_car_parking_spaces     False
total_of_special_requests       False
reservation_status              False
reservation_status_date         False
dtype: bool
```

2. 基本情况：城市酒店和假日酒店预订需求和入住率比较

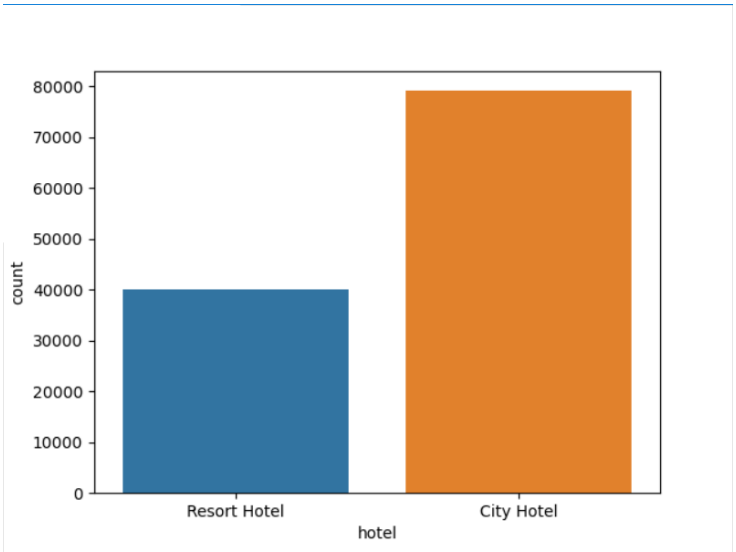
```
## 1. 基本情况：城市酒店和假日酒店预订需求和入住率比较
print("df.columns:\n", df.columns)
print("df.hotel.value_counts():\n", df.hotel.value_counts())

sns.countplot(df.hotel)
plt.show()
city_count_book = df.hotel.value_counts()[0] # 城市酒店的预定情况
resort_count_book = df.hotel.value_counts()[1] # 假日酒店的预定情况

city_check_in = df[df['hotel'] == 'City Hotel'].is_canceled.value_counts()[0] # 城市酒店取消预订情况
resort_check_in = df[df['hotel'] == 'Resort Hotel'].is_canceled.value_counts() # 假日酒店取消预定情况

# 入住率=入住总数/预定总数
city_rate = city_check_in/city_count_book
resort_rate = resort_check_in/resort_count_book

print('城市酒店入住率：{}，假日酒店入住率：{}'.format(city_rate, resort_rate))
```




```
df.columns:
 Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
       'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',
       'country', 'market_segment', 'distribution_channel',
       'is_repeated_guest', 'previous_cancellations',
       'previous_bookings_not_canceled', 'reserved_room_type',
       'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',
       'company', 'days_in_waiting_list', 'customer_type', 'adr',
       'required_car_parking_spaces', 'total_of_special_requests',
       'reservation_status', 'reservation_status_date'],
      dtype='object')
df.hotel.value_counts():
 City Hotel      79163
Resort Hotel    40047
Name: hotel, dtype: int64
城市酒店入住率：0.5821406465141543，假日酒店入住率：0    0.722326
```

分析可知，就酒店预定来说城市酒店比假日酒店更受欢迎，人们更喜欢预定城市酒店，但是假日酒店的酒店入住率更高。

3. 用户行为：提前预订时间、入住时长、餐食预订情况

3.1 提前预订时间

```
均值：104.10922741380756
中位数：  69.0
最小值：  0
最大值：  737
四分位数：[ 18.  69. 161.]
众数：  0
```

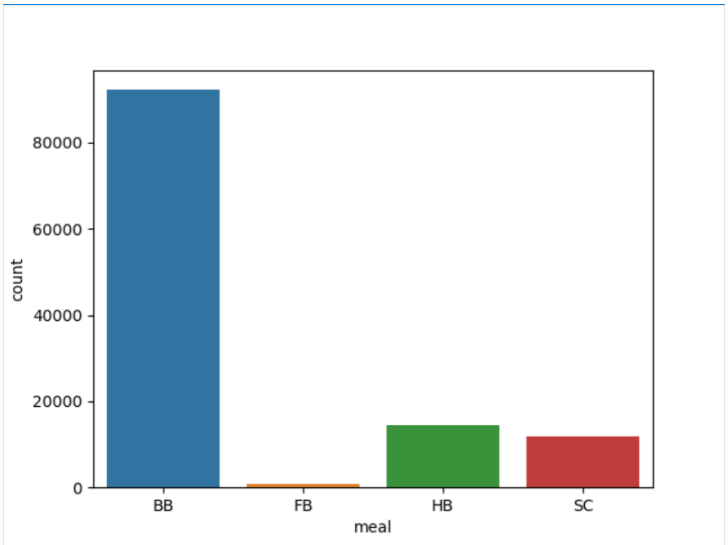
顾客平均提前预定时间为 104 天左右，预定最久的天数为 737 天，将近两年多。大部分顾客都是当天预定当天入住

3.2 入住时长

```
均值: 3.4262477980035233
中位数:  3.0
最小值:  0
最大值:  69
四分位数: [2. 3. 4.]
众数:  2
```

顾客平均入住晚数为 3 晚左右，最大入住晚数为 69 天，两个多月，其中大部分顾客入住 2 晚。

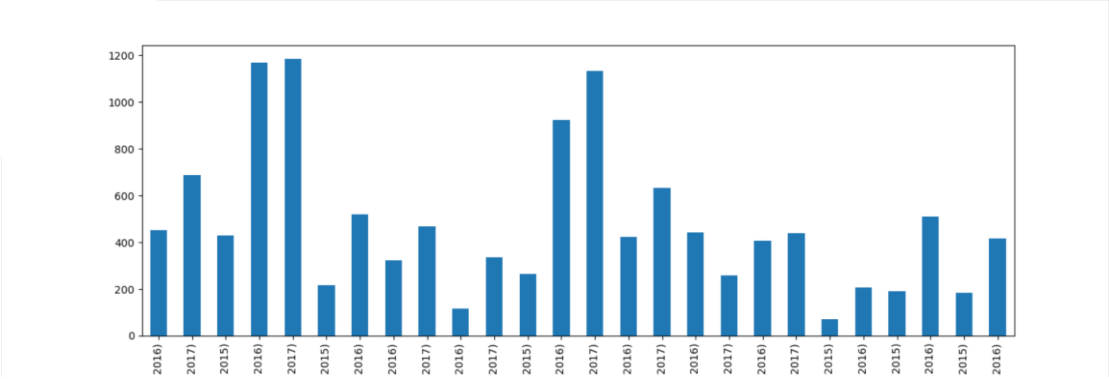
3.3 餐食预订情况



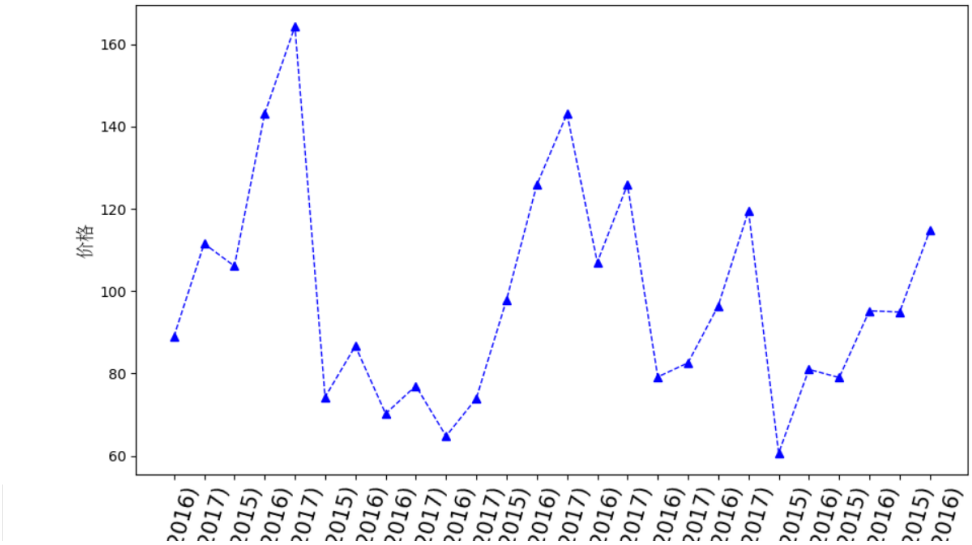
大多数人会在酒店中订餐，其中大部分人预定了 BB 这个套餐类型，很少人订 FB 这个套餐类型，少部分人不需要订餐服务（SC）

4. 一年中最佳预订酒店时间

4.1 酒店入住情况柱状图



4.2 酒店平均价格-时间段折线图



结合上面两个图来看，对于顾客来说，最佳预定酒店的时间应为每年的 1、2 月和 11、12 月，这几个时间段的酒店的入住人数少且价格较低，是最佳的酒店预定入住时间。

4. 利用 Logistic 预测酒店预订

```
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\Administrator\envs\numpy\lib\site-packages\sklearn\linear_model\_logistic.py:940: Co
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
LR_model cross validation accuarcy score: 0.7947 +/- 0.0027 (std) min: 0.7915, max: 0.7984
PS C:\Users\Administrator\Desktop\作业\数据挖掘\第三次互评作业\homework3>
```

用逻辑回归来预测的准确率结果如上图倒数第二行所示，在 79%左右。