EXAM 2024

Business Analytics with Python – Pricing and Revenue Management

For coding questions, you are expected to submit a functioning python file relying only packages introduced in the lecture. Code comments will help ensure partial credits if functionalities are incomplete or incorrect.

The code stumps for the questions can be found in the following github assignment:

https://classroom.github.com/a/IE3JJQXR

Question 1: Profit Optimization for Hotel Room Sales (15 points)

A hotel wants to optimize the revenue from its room bookings. The hotel has a total of 200 rooms. The variable cost to maintain each room per night varies with occupancy:

- $50 per room for up to 60 rooms.
- $40 per room for more than 60 rooms and up to 150 rooms.
- $30 per room for more than 150 rooms.

Additionally, if the number of bookings exceeds 100 rooms, the hotel incurs an extra administrative fee of $20,000.

The demand function for room bookings is given by $D(p) = \lfloor 1000 - 2.2p \rfloor$, where p represents the selling price (in dollars) of each room per night. The floor function $\lfloor \cdot \rfloor$ ensures integer demand levels by rounding down (you can achieve this by casting to (int) in Python).

Tasks:

a. **Formulate the Expressions and Implement in Python**

1. **Demand Function:** Implement a Python function `demand(p)` to calculate the number of rooms booked based on the selling price.
2. **Revenue Function:** Implement a Python function `revenue(p)` to calculate the total revenue.
3. **Total Cost Function:** Implement a Python function `total_cost(p)` to calculate the total cost, considering the variable cost tiers and the extra administrative fee if bookings exceed 100 rooms.

b. **Profit Calculation** Implement a Python function `profit(p)` to calculate the profit based on the total revenue and total cost.

Question 2: Hotel Reservation Management (15 points)

The reservation system of a hotel needs to manage room bookings efficiently. The hotel operates a price-based booking system which determines whether to accept a booking based on the ADR. Reservations specify the number of rooms needed and an ADR (Average Daily Rate). The system should determine based on simple rules whether the request is served and update the data accordingly.

Tasks:

a. **Single Reservation Handling** Implement a Python function `process_reservation(hotel, reservation)` that returns the hotel data object after processing the. The function should follow the following rules to determine whether or not a booking is accepted:

- The reservation's ADR must be greater than or equal to the class's ADR threshold.
- The capacity must be sufficient to accommodate the entire reservation.

When these requirements are cleared the booking is entered into the bookings list and the number of available rooms is updated accordingly.

The `hotel` dictionary should contain:

- `rooms_available` (number of rooms available to both booking classes)
- `adr_threshold` (minimum acceptable ADR)
- `bookings` (list of accepted bookings)

The `reservation` dictionary should contain:

- `rooms` (number of rooms requested)
- `adr` (Average Daily Rate of the reservation)
- `id` (String for identification)

b. **Process Multiple Reservations** Extend the previous function to handle a list of reservations. Implement a Python function `process_reservations(hotel, reservations)` that returns the bookings list of the hotel.

HINT: You will want to call your previously defined function `process_reservation(hotel, reservation)` to do this!

Question 3: Demand Function Analysis with Pandas (20 points)

You are provided with a synthetic dataset `booking-data.csv` of hotel booking requests. Each request represents one customer, specifying the ADR, room type, number of rooms, and location. The data can be found at:

https://raw.githubusercontent.com/chrisflath/MPS_2019/master/booking-data.csv

Alternatively it is provided in the exam github repository.

Tasks:

a. **Descriptive Statistics and Data Exploration**

- Load the csv file dataset into a Pandas DataFrame.
- Perform basic exploratory data analysis (EDA) to understand the dataset's structure and main characteristics. This should include:
    - Descriptive statistics (mean, median, standard deviation, etc.) for numerical columns.
    - Distribution of different room types and locations.
    - Any interesting trends or patterns over time.

- Group the data by `location` and `room_type` and present grouped summaries in a clear and concise format.

b. **Demand Function Analysis and Plotting**

- Define a `demand` function that calculates the total number of rooms booked for a given location, ADR, and room type.
    - **Hint**: The function should filter the dataset for rows that match the specified location and room type, and where the ADR is greater than or equal to the specified value. Then sum the `num_rooms` column.
- Populate a DataFrame with the demand values for a range of ADR values for each location and room type.
    - **Hint**: Evaluate the demand function for a range of ADR values (e.g., from 50 to 600) and store the results in a DataFrame.
- Plot the demand function D(p) that represents the number of rooms booked (`num_rooms`) as a function of the ADR for a continuum of ADR values.
- Create separate plots for each hotel location with different colors representing different room types.

c. **Reflection**

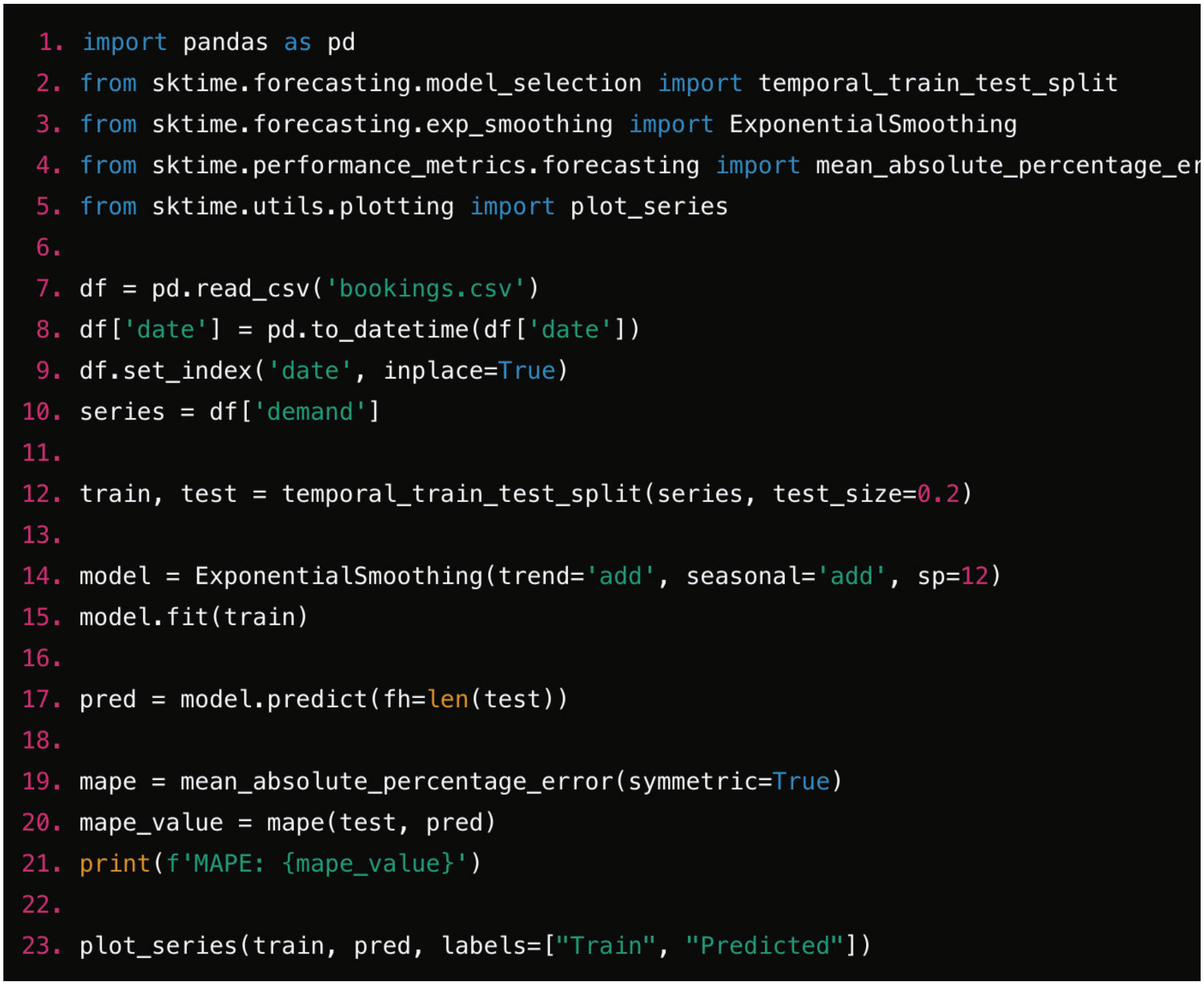
The above analysis rested on a generated data set. Explain why it may be difficult to obtain such data in the real world. Suggest a possible approach and describe some details. (approx.150 words)

Question 4: Forecasting (20 points)

You are provided with a description of a dataset and a Python script for time series forecasting using historical booking data with the `sktime` library. The dataset contains historical booking data for a hotel, with the following schema:

- `date`: The date of the booking (format: YYYY-MM-DD)
- `demand`: The number of rooms booked on that date
- `holiday_flag`: Indicates whether the date is a holiday (1 for holiday, 0 otherwise)
- `special_events`: Indicates whether there are special events occurring near the hotel on that date (1 for special event, 0 otherwise)

The script is provided as a screenshot below – do not implement or try to run it!



```python
import pandas as pd
from sktime.forecasting.model_selection import temporal_train_test_split
from sktime.forecasting.exp_smoothing import ExponentialSmoothing
from sktime.performance_metrics.forecasting import mean_absolute_percentage_er
from sktime.utils.plotting import plot_series

df = pd.read_csv('bookings.csv')
df['date'] = pd.to_datetime(df['date'])
df.set_index('date', inplace=True)
series = df['demand']

train, test = temporal_train_test_split(series, test_size=0.2)

model = ExponentialSmoothing(trend='add', seasonal='add', sp=12)
model.fit(train)

pred = model.predict(fh=len(test))

mape = mean_absolute_percentage_error(symmetric=True)
mape_value = mape(test, pred)
print(f'MAPE: {mape_value}')

plot_series(train, pred, labels=["Train", "Predicted"])
```

a. Explaining the Script

- Explain in a detailed manner (not only the function call but also the arguments and the rationale) the following lines of the code:
    - 9
    - 12
    - 14
    - 15
    - 17
    - 19,20
    - 23

b. Towards causal forecasting

- Discuss why the above model cannot readily incorporate external features like `holiday_flag` and `special_events` in its fit. You may want to use the underlying mathematical formulation to explain this issue.
- Explain the limitations this imposes on the forecasting model.

c. Alternative Prediction Approach

- Propose an alternative approach to incorporate external features such as `holiday_flag` and `special_events` into the demand forecasting model. Name the relevant packages and a useful algorithm for this approach.
- Describe the feature engineering necessary to match the expressiveness of the time series model.

Question 5: Spa Pricing Optimization (20 points)

A spa wants to optimize its revenue by determining the best price levels for its treatments while considering resource constraints. The spa offers three types of treatments: Basic Massage, Aromatherapy, and Deep Tissue Massage. Each treatment requires different amounts of resources (e.g., time from specialists), and the spa has a limited total resource capacity.

You are provided with the following code that formulates and solves the optimization problem using the `PuLP` library. The code aims to find the optimal price levels for each treatment type that maximize total profit while respecting the resource constraints.

```
try:
  from pulp import *
except:
  !pip install pulp
  from pulp import *

import numpy as np


# Example data
treatments = ['Basic Massage', 'Aromatherapy', 'Deep Tissue Massage']
price_range = np.arange(10, 101, 10)  # Pricing levels
a_values = {'Basic Massage': 1500, 'Aromatherapy': 2000, 'Deep Tissue Massage': 1600
b_values = {'Basic Massage': 1.5, 'Aromatherapy': 1.2, 'Deep Tissue Massage': 1.3}
resource_usage_per_unit = {'Basic Massage': 1, 'Aromatherapy': 2, 'Deep Tissue Massage': 3}
total_resource_capacity = 50 # Total resource capacity (e.g., hours available from specialists)

demand = {}
for treatment in treatments:
    demand[treatment] = {}
    for p in price_range:
        demand[treatment][p] = max(a_values[treatment] * p**(-b_values[treatment]), 0)

prob = LpProblem("Spa_Treatment_Profit_Maximization", LpMaximize)

price_vars = {}
for treatment in treatments:
  price_vars[treatment] = LpVariable.dicts(treatment+"price", price_range, cat='Binary')

prob += lpSum(price_vars[treatment][price] * demand[treatment][price] * price for treatment in
treatments for price in price_range)

# Constraint Block 1
for treatment in treatments:
    prob += lpSum(price_vars[treatment][price] for price in price_range) == 1

# Constraint Block 2
prob += lpSum(price_vars[treatment][price] * demand[treatment][price] *
resource_usage_per_unit[treatment]
                   for treatment in treatments for price in price_range) <=
total_resource_capacity

# Solve the problem
prob.solve(PULP_CBC_CMD(msg=False))
```

a. **Specify and Explain the Demand Functions**

- Briefly explain how the example demand data is populated – what is the demand function? Which price levels are considered?
- Explain why the chosen demand functions cannot be used directly in the revenue term of a linear programming model. (Remark: There are two reasons.)

b. **Explain the Optimization Logic**

Describe the decision variables, objective function, and constraints used in the optimization model.

c. **Determine Optimal Prices and Profits**

Use the provided code to determine the optimal prices, profit, and demand levels for a resource capacity of 30 units.

d. **Resource Capacity Sensitivity Analysis**

- Extend the code to evaluate the profit for different resource levels ranging from 20 to 100 in increments of 10.
- Plot the total profit and demand for each treatment against the resource capacity.