Studio Shodwe

Predictive Insights for Enhanced Booking Performance: A Regression-Based Approach

By: Muhammad Arkhan Doohan DS25

Table Of Content

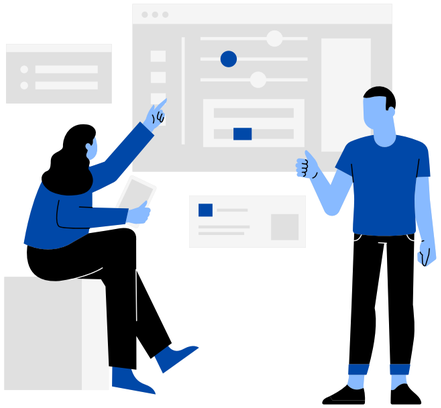
01 Data Preparation

02 EDA

03 Modeling

04 Evaluation

Data Pre-Processing



Latarbelakang : Data yang akan dianalisa berkaitan dengan data reservation Hotel. Melalui data ini nantinya diharapkan bisa kita prediksi sebetulnya apa yang membuat costumer pengunjung hotel-hotel untuk bisa closing dan booking kamar, dan apa penyebab mereka cancel booked.

Output: Melalui analisa ini diharapkan dapat memberikan rekomendasi dari sisi mana perhotelan dapat mengambil langkah efektif dalam melakukan promo dengan segmentasi yang tepat.

Saya meload dataset dengan membaca google drive dan mencari lokasi dataset yang sudah disediakan

Load Data

```
from google.colab import drive
drive.mount('/content/drive')

import pandas as pd
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
# Ganti path dengan path file di Google Drive Anda
df = pd.read_csv('/content/drive/My Drive/Dibimbing/InterviewU2/train.csv')
df.head()
```

	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_we
0	Crystal Cove, Barbados Barbados	0	8	2019	January	2	10	0	
1	Greensboro Courtyard Greensboro, NC	1	524	2018	December	51	15	0	

Load Data Set | **Data Understanding** | **Data Cleaning** | **Feature Engineering**

Data Pre-Processing

```
hotel: Type of hotel (Categorical)

is_canceled: Whether the booking was canceled or not (Binary

lead_time: Number of days between booking date and arrival d

arrival_date_year: The year of the arrival date (Numeric)

arrival_date_month: The month of the arrival date (Categoric

arrival_date_week_number: The week number of the arrival dat

arrival_date_day_of_month: The day of the month of the arriv

stays_in_weekend_nights: Number of weekend nights stayed or

stays_in_week_nights: Number of week nights stayed or booked

adults, children, babies: Number of guests categorized by ag
```

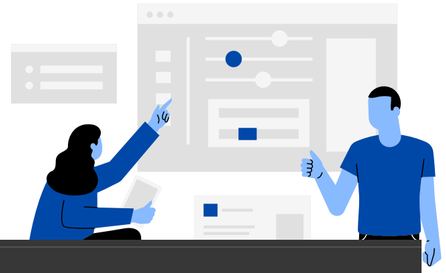
Business Understanding : Bagian ini yaitu mencoba untuk memahami maksud dari data, memahami apa saja makna di dalam kolom-kolomnya, maka data analyst dapat memposisikan dirinya sebagai siapa, dan bisa memperkirakan insight apa yang di dapat dengan kelengkapan data tersebut.

Selain mengetahui secara bisnis, bagian ini juga untuk mengetahui gambaran umum dari data, seperti type data, jumlah kolom dan baris. dsb

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 83293 entries, 0 to 83292
Data columns (total 33 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   hotel                           83293 non-null  object
 1   is_canceled                     83293 non-null  int64
 2   lead_time                       83293 non-null  int64
 3   arrival_date_year               83293 non-null  int64
 4   arrival_date_month              83293 non-null  object
 5   arrival_date_week_number        83293 non-null  int64
 6   arrival_date_day_of_month       83293 non-null  int64
 7   stays_in_weekend_nights         83293 non-null  int64
 8   stays_in_week_nights            83293 non-null  int64
 9   adults                          83293 non-null  int64
 10  children                        83290 non-null  float64
 11  babies                          83293 non-null  int64
 12  meal                            83293 non-null  object
 13  country                         82947 non-null  object
 14  market_segment                  83293 non-null  object
 15  distribution_channel            83293 non-null  object
 16  is_repeated_guest               83293 non-null  int64
 17  previous_cancellations          83293 non-null  int64
 18  previous_bookings_not_canceled  83293 non-null  int64
 19  reserved_room_type              83293 non-null  object
```

Load Data Set | **Data Understanding** | Data Cleaning | Feature Engineering

Data Pre-Processing



pada bagian data cleaning ini mencoba memebersihkan data yang duplikat, data yang missing value maupun jika terdapat data yang outlier. Untuk treatment nya juga berbeda-beda di setiap kondisinya.

Pada case data ini, tidak ada data yang **duplikat** namun terdapat beberapa kolom yang missing value yaitu children (**3**), country (**346**), agent (**11,404**), dan company (**78,559**) dari **83,293** baris. Ada beberapa treatment yang dilakukan untuk semua missing value tersebut.

- Kolom **children** karena yang missing value itu hanya 3 baris jadi saya lebih kepada **menghapus baris** tersebut
- Kolom **company** karena yang missing sudah betul-betul lebih dari **50%** kosong maka metode yang saya gunakan adalah **menghapus kolom**.
- Terakhir untuk kolom **agent** dan **company**, meskipun dua-duanya memiliki missing value yang tidak terlalu banyak atau masih di bawah **50%**. **Awalnya saya ingin memasukan median/mean** pada data tersebut dengan melihat outliernya. Namun, setelah saya telaah dan pahami datanya. kolom tersebut yaitu menjelaskan tentang entitas yaitu agent dan nama perusahaan, yang mana untuk entitas tidak sembarangan kita langsung isi saja, karena akan berpengaruh besar terhadap data. Sehingga saya lebih memilih **menghapus barisnya**.

```
df.isnull().sum()

hotel                             0
is_canceled                       0
lead_time                         0
arrival_date_year                 0
arrival_date_month                0
arrival_date_week_number          0
arrival_date_day_of_month         0
stays_in_weekend_nights           0
stays_in_week_nights              0
adults                            0
children                          3
babies                            0
meal                              0
country                         346
market_segment                    0
distribution_channel              0
is_repeated_guest                 0
previous_cancellations            0
previous_bookings_not_canceled    0
reserved_room_type                0
assigned_room_type                0
booking_changes                   0
deposit_type                      0
agent                         11404
company                       78559
days_in_waiting_list              0
customer_type                     0
adr                               0
required_car_parking_spaces       0
```

```
+ Code  + Text

[17] df = df.drop('company', axis=1)

[18] df = df.dropna(subset=['country', 'agent'])

[20] df = df.dropna(subset=['children'])

df.isnull().sum()

hotel                             0
is_canceled                       0
lead_time                         0
arrival_date_year                 0
arrival_date_month                0
arrival_date_week_number          0
arrival_date_day_of_month         0
stays_in_weekend_nights           0
stays_in_week_nights              0
adults                            0
children                          0
babies                            0
meal                              0
country                           0
market_segment                    0
distribution_channel              0
```

Load Data Set | Data Understanding | Data Cleaning | Feature Engineering

Data Pre-Processing

Feature Engineering : Pada bagian ini saya mencoba untuk membuat kolom baru yaitu total visitors, atau total pengunjung. mengingat pada kolom itu terdapat 3 kolom yang termasuk sebagai pengunjung yaitu Adult (Dewasa), Children (Anak) dan Babies (Bayi). Saya coba gabungkan itu semua dengan sebutan total visitors

Feature Engineering

```
df[['adults', 'children', 'babies', 'meal']]
```

Show hidden output

```
[52] # Mengubah tipe data kolom 'children' menjadi integer
     df['children'] = df['children'].astype(int)
```

```
[53] # Membuat kolom baru 'total_visitors'
     df['total_visitors'] = df['adults'] + df['children'] + df['babies']
```

```
df.head()
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_n
0	Crystal Cove, Barbados Barbados	0	8	2019	January	
1	Greensboro Courtyard Greensboro, NC	1	524	2018	December	

Feature Engineering

```
df[['adults', 'children', 'babies', 'meal']]
```

	adults	children	babies	meal
0	2	0.0	0	BB
1	2	0.0	0	BB
2	2	0.0	0	BB
4	2	0.0	0	BB
5	2	0.0	0	BB
...	...	...	...	...
83287	2	0.0	0	BB
83288	2	1.0	0	BB
83289	2	0.0	0	BB
83291	2	1.0	0	BB
83292	2	0.0	0	BB

71771 rows × 4 columns

Load Data Set | Data Understanding | Data Cleaning | **Feature Engineering**

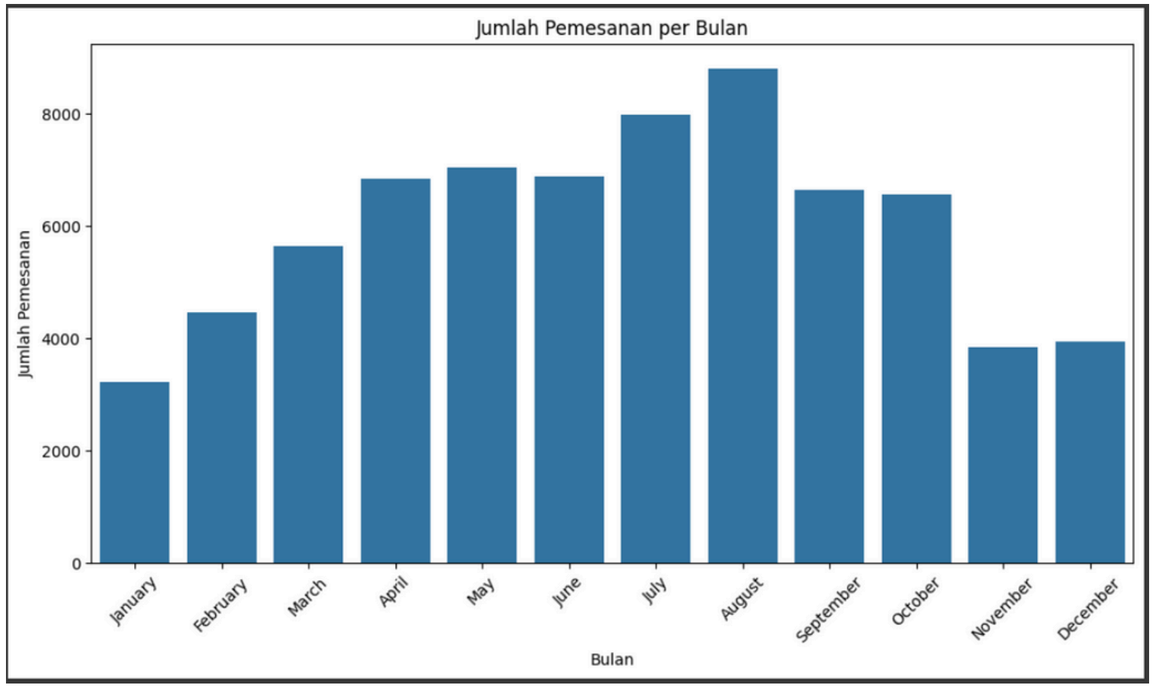
Exploratory Data Analysis (EDA)

- Pegunjung paling banyak pada bulan apa
- Korelasi Tertinggi dengan Canceled Booked
- Lama waktu Pemesanan (Time Lead) dengan pembatalan apakah memiliki hubungan?

Exploratory Data Analysis (EDA)

EDA 1: saya ingin coba melihat pada bulan mana jumlah pemesanan hotel itu terbanyak, dilihat bahwa terbanyak ada pada pertengahan tahun, d bulan agustus. Namun ini juga sejalan dengan pengcancel-an yang terjadi banyak pada bulan agustus juga dengan **3,464** cancel booked. Oleh karena itu melalui model ini kita akan menjadi lebih bersiap bulan kapan kita harus lebih menguatkan efisiensi dan pelayanan.

Pegunjung paling banyak pada bulan apa




```python
# Menghitung jumlah pembatalan per bulan
cancelations_per_month = df[df['is_canceled'] == 1]['arrival_date_month'].value_counts()

# Menemukan bulan dengan pembatalan tertinggi
month_with_highest_cancelations = cancelations_per_month.idxmax()
number_of_cancelations = cancelations_per_month.max()

print(f"Bulan dengan pembatalan tertinggi adalah {month_with_highest_cancelations} dengan {number_of_c
```

```
Bulan dengan pembatalan tertinggi adalah August dengan 3464 pembatalan.
```

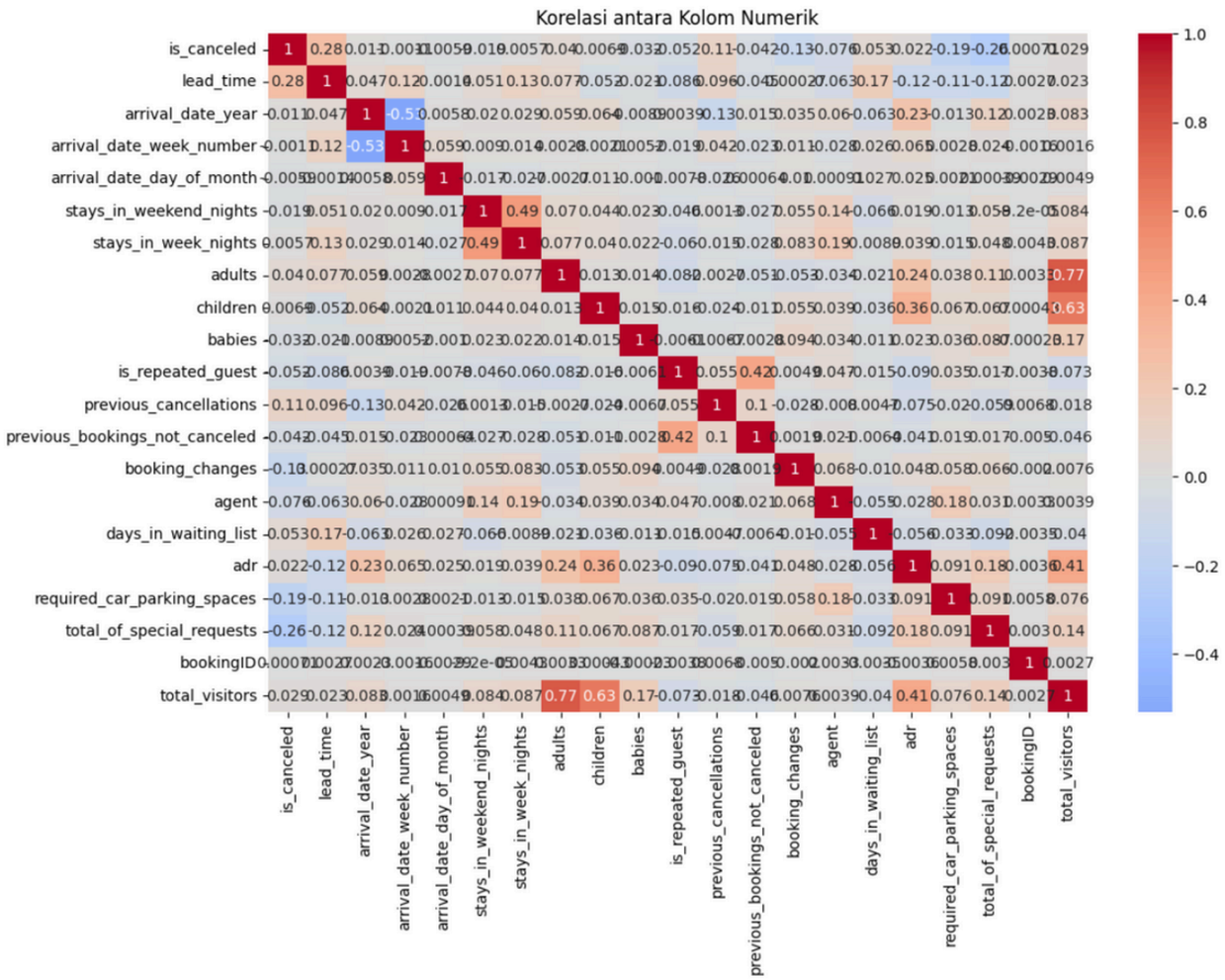
Exploratory Data Analysis (EDA)

Korelasi Tertinggi dengan Canceled Booked

Untuk melihat korelasi dari setiap kolom-kolom saya masih tertarik untuk menggunakan heatmap, namun dikarenakan kolom yang sangat banyak, sehingga tidak begitu jelas untuk korelasinya. Solusinya saya membuat nilai korelasi tersebut melalui list, dengan diurutkan tertinggi berdasarkan kolom target (cancel booked). Maka terlihat untuk korelasi tertinggi ada pada **lead_time**.

```
[33] correlation_with_canceled = correlation_matrix['is_canceled'].sort_values(ascending=False)
     print(correlation_with_canceled)

is_canceled                     1.000000
lead_time                       0.275061
previous_cancellations          0.113713
days_in_waiting_list            0.052720
adults                          0.039560
total_visitors                  0.029356
adr                             0.021664
arrival_date_year               0.010800
children                        0.006929
stays_in_week_nights            0.005687
bookingID                       0.000713
arrival_date_week_number       -0.001076
arrival_date_day_of_month      -0.005940
stays_in_weekend_nights        -0.019238
babies                         -0.032453
previous_bookings_not_canceled -0.042480
is_repeated_guest              -0.051703
agent                          -0.075826
booking_changes                -0.130718
required_car_parking_spaces    -0.186356
total_of_special_requests      -0.256487
Name: is_canceled, dtype: float64
```

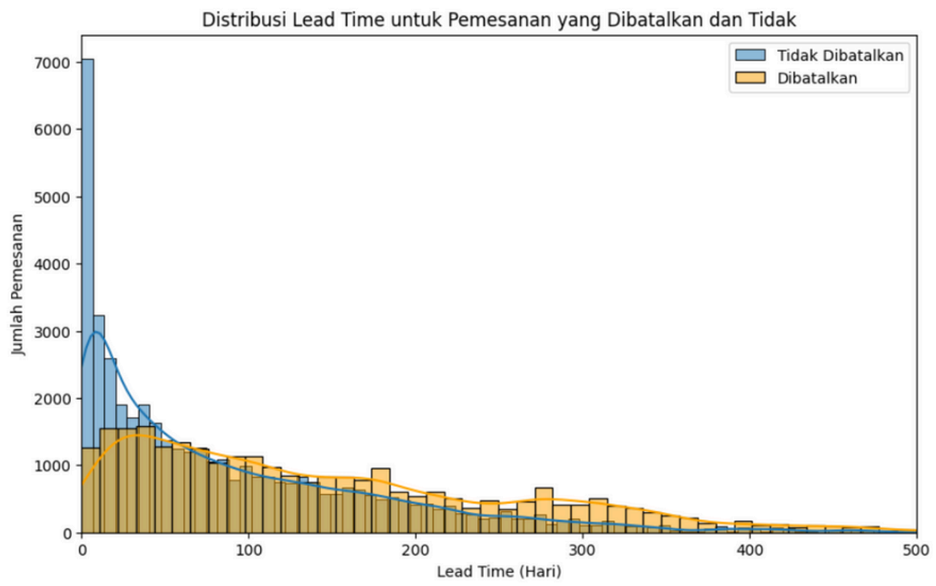


Exploratory Data Analysis (EDA)

Lama waktu Pemesanan (Time Lead) dengan pembatalan apakah memiliki hubungan?

Melalui eda sebelumnya bahwa lead_time menjadi kolom yang paling berkolerasi. saya coba untuk menganalisa lebih dalam. Seberapa lama rata-rata hari orang yang akhirnya mengcancel kamar yaitu di 56 hari atau 2 bulanan. Oleh karena itu ita dapatkan insight bahwa:

Karena total pengunjung paling banyak itu di bulan agustus, dan pemesanan paling sering itu di 56 hari sebelum atau 2 bulan sebelum yaitu bulan Juni. Maka kita bisa menentukan waktu yang tepat untuk meningkatkan pelayanan dan meningkatkan promo pada bulan Juni-Agustus.




```
[38] # Mencari lead time yang paling sering dibatalkan
     lead_time_canceled = df[df['is_canceled'] == 1]['lead_time'].value_counts().idxmax()
     print("Lead time yang paling sering dibatalkan:", lead_time_canceled, "hari")

     Lead time yang paling sering dibatalkan: 56 hari

     # Menghitung statistik deskriptif untuk lead time berdasarkan status pembatalan
     print(df.groupby('is_canceled')['lead_time'].describe())

                   count        mean         std  min   25%    50%    75%    max
     is_canceled
     0            43848.0   87.996237   91.973877  0.0  15.0   57.0  135.0  542.0
     1            27923.0  148.771013  119.525435  0.0  53.0  117.0  219.0  629.0
```

Pemodelan Mechine Learning

- Regrestion
- Evaluation

Modeling - Regression Method

Pemodelan

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
```

```python
# Pilih fitur-fitur yang akan digunakan dalam model
features = ['lead_time', 'arrival_date_week_number', 'arrival_date_day_of_month',
            'stays_in_weekend_nights', 'stays_in_week_nights', 'adults', 'children',
            'babies', 'is_repeated_guest', 'previous_cancellations',
            'previous_bookings_not_canceled', 'booking_changes', 'agent',
            'days_in_waiting_list', 'adr', 'required_car_parking_spaces',
            'total_of_special_requests','total_visitors']
```

```python
# Ubah fitur kategorikal menjadi numerik menggunakan one-hot encoding
df_encoded = pd.get_dummies(df[features],
                            columns=['agent'])
```

```python
# Bagi data menjadi data latih dan data uji
X = df_encoded
y = df['is_canceled']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Buat model regresi logistik
model = LogisticRegression()
```

```python
# Latih model
model.fit(X_train, y_train)
```

```python
# Buat model regresi logistik
model = LogisticRegression()
```

```python
# Latih model
model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
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
  n_iter_i = _check_optimize_result(
```

```
▾ LogisticRegression
LogisticRegression()
```

```python
# Prediksi pada data uji
y_pred = model.predict(X_test)
```

Alasan Menggunakan Regresion karena ini termasuk data yang supervised dan numerical yaitu antara cancel atau tidak cancel. Antara 1 dan 0 maka model mechine learning yang cocok dengan akurasi yang paling bagus dengan karakter seperti itu adalah model regresi.

Untuk prosesnya sendiri yaitu pertama dengan menentukan feature-feature mana saja yang sama-sama numerical untuk di masukan ke dalam uji pemodelan. Jika terdapat kolom yang belum numerical maka bisa di encoding terlebih dahulu menjadi numerical. Setelah itu baru di buat pemodelannya menggunakan sklearn.

Evaluation Model

Evaluasi Model

```python
# Evaluasi model
accuracy = accuracy_score(y_test, y_pred)
print("Akurasi:", accuracy)

print(classification_report(y_test, y_pred))
```

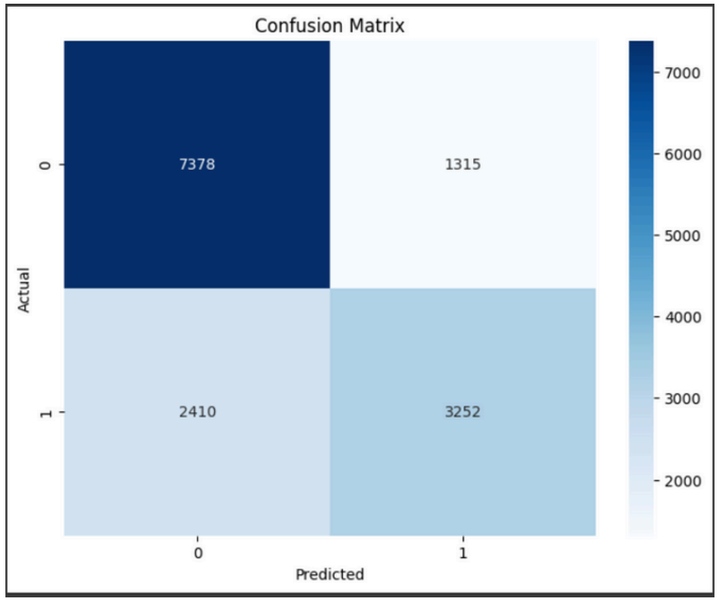
```
Akurasi: 0.7405085336119819
              precision    recall  f1-score   support

           0       0.75      0.85      0.80      8693
           1       0.71      0.57      0.64      5662

    accuracy                           0.74     14355
   macro avg       0.73      0.71      0.72     14355
weighted avg       0.74      0.74      0.73     14355
```

```python
[71] from sklearn.metrics import confusion_matrix, roc_curve, auc, precision_recall_curve
     import matplotlib.pyplot as plt
```

```python
[72] # Confusion Matrix
     cm = confusion_matrix(y_test, y_pred)
     plt.figure(figsize=(8, 6))
     sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
     plt.title('Confusion Matrix')
     plt.xlabel('Predicted')
     plt.ylabel('Actual')
     plt.show()
```



Setelah membuat pemodelan machine learning. Maka yang dilakukan selanjutnya yaitu melakukan model evaluasi, untuk melihat apakah sudah bagus atau belum secara akurasi pemodelannya. Dihasilkan bahwa hasilnya adalah

- precision nya diangka 75%
- F1 Scorenya 80%
- dan recal nya 85%

yang mana artinya pemodelan ini bisa dibilang sudah bagus dan akurat diatas 70%. Saya masih penasaran dan mencoba membuktikannya menggunakan confusion matrix hasilnya bahwa hasil prediksinya banyak betulnya, yang diprediksi betul lebih banyak yang diprediksi salah 7378(prediksi benar booked) dan 3235 (prediksi benar cancel)

Thanks