

Holiday Package Prediction

Dokumen
Laporan Final
Project



Kelompok 3 – Pandas Lovers

Ketua : Sendhy Boedhi

Anggota :

1. Edgar Ariel Majied
2. Vionella Awanda Irsabadi
3. Teguh Ferdianto
4. R. Arnanda Adi Wijanarko
5. Faris Isham Wiransyah
6. Jodhi Krisantus Sihalbu
7. Jannisah Dwi Rahhadiski

Stage 0

PREPARATION

Latar Belakang Masalah

1. Problem Statement

Perusahaan traveling 'Trips & Travel' membuat penawaran paket liburan terbaru. Agar lebih efektif, perusahaan ingin menyelesaikan permasalahan yang ada.

- Pada tahun lalu, hanya 19% pelanggan yang membeli paket liburan yang ditawarkan.
- *Revenue* perusahaan tidak mengalami peningkatan yang signifikan.

2. Role

PT. Pandas Lovers merupakan perusahaan yang bergerak di bidang konsultasi bisnis. Perusahaan kami bertugas untuk menganalisis bisnis, menciptakan solusi, membantu, dan mengembangkan rencana bisnis yang efisien. Perusahaan menggunakan data-data klien yang kemudian dikembangkan menjadi sebuah wawasan baru untuk memenuhi tujuan para klien.

Berikut anggota tim data beserta perannya.

Nama	Peran
Sendhy	<i>Project Manager</i>
Edgar	<i>Data Scientist</i>
Vionella	<i>Data Analyst</i>
Faris	<i>Deployment Specialist</i>

Nama	Peran
Arnanda	<i>Business Analyst</i>
Jannisah	<i>Model Optimization Specialist</i>
Teguh	<i>Data Engineer</i>
Jodhi	<i>Machine Learning Engineer</i>

3. Goal

Untuk menaikkan *conversion rate* dan *revenue* perusahaan.

4. Objective

Membuat model untuk memprediksi pelanggan yang akan membeli paket liburan terbaru.

5. Business Metrics

Total *revenue* dan *conversion rate*.

Stage 1

EDA, INSIGHT, & VISUALIZATION



Descriptive Statistics

1A. Kolom dengan tipe data yang kurang sesuai

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4888 entries, 0 to 4887
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CustomerID      4888 non-null   int64  
 1   ProdTaken        4888 non-null   int64  
 2   Age              4662 non-null   float64 
 3   TypeofContact    4863 non-null   object  
 4   CityTier         4888 non-null   int64  
 5   DurationOfPitch 4637 non-null   float64 
 6   Occupation       4888 non-null   object  
 7   Gender            4888 non-null   object  
 8   NumberOfPersonVisiting 4888 non-null   int64  
 9   NumberOfFollowups 4843 non-null   float64 
 10  ProductPitched   4888 non-null   object  
 11  PreferredPropertyStar 4862 non-null   float64 
 12  MaritalStatus     4888 non-null   object  
 13  NumberOfTrips     4748 non-null   float64 
 14  Passport          4888 non-null   int64  
 15  PitchSatisfactionScore 4888 non-null   int64  
 16  OwnCar            4888 non-null   int64  
 17  NumberOfChildrenVisiting 4822 non-null   float64 
 18  Designation        4888 non-null   object  
 19  MonthlyIncome      4655 non-null   float64 

dtypes: float64(7), int64(7), object(6)
memory usage: 763.9+ KB
```

Self Enquiry	3444
Company Invited	1419
Name: TypeofContact, dtype: int64	
Salaried	2368
Small Business	2084
Large Business	434
Free Lancer	2
Name: Occupation, dtype: int64	
Male	2916
Female	1817
Fe Male	155
Name: Gender, dtype: int64	
Basic	1842
Deluxe	1732
Standard	742
Super Deluxe	342
King	230
Name: ProductPitched, dtype: int64	
Married	2340
Divorced	950
Single	916
Unmarried	682
Name: MaritalStatus, dtype: int64	
Executive	1842
Manager	1732
Senior Manager	742
AVP	342
VP	230
Name: Designation, dtype: int64	

- Terdapat 6 kolom dengan tipe data yang kurang sesuai, yaitu kolom *NumberOfChildrenVisiting*, *DurationOfPitch*, *NumberOfFollowups*, *PreferredPropertyStar*, *NumberOfTrips*, dan *Age* yang bertipe data float, sehingga nantinya harus kita ubah menjadi tipe data integer.
- Terdapat kesalahan value pada kolom *Gender* yaitu adanya value 'Fe Male', yang nantinya harus kita handle pada saat pre-processing data.
- Pada kolom *MaritalStatus*, terdapat value yang memiliki arti sama yaitu value 'Divorced', 'Single', dan 'Unmarried'.

Tabel Data Duplicate dan Missing Value

	Missing Value Total	Missing Value Percentage (%)
CustomerID	0	0.000000
ProdTaken	0	0.000000
Age	226	4.623568
TypeofContact	25	0.511457
CityTier	0	0.000000
DurationOfPitch	251	5.135025
Occupation	0	0.000000
Gender	0	0.000000
NumberOfPersonVisiting	0	0.000000
NumberOfFollowups	45	0.920622
ProductPitched	0	0.000000
PreferredPropertyStar	26	0.531915
MaritalStatus	0	0.000000
NumberOfTrips	140	2.864157
Passport	0	0.000000
PitchSatisfactionScore	0	0.000000
OwnCar	0	0.000000
NumberOfChildrenVisiting	66	1.350245
Designation	0	0.000000
MonthlyIncome	233	4.766776

1B. Kolom yang memiliki nilai kosong

- Terdapat 8 kolom yang memiliki missing value, yaitu kolom *Age*, *TypeofContact*, *DurationOfPitch*, *NumberOfFollowups*, *PreferredPropertyStar*, *NumberOfTrips*, *NumberOfChildrenVisiting*, dan *MonthlyIncome*. Kolom-kolom tersebut memiliki missing value < 10%, sehingga dapat dikategorikan masih relatif aman.
- Tidak terdapat data yang duplikasi pada dataset ini.

Tabel Describe Kolom Diskrit

	count	mean	std	min	25%	50%	75%	max
ProdTaken	4888.0	0.188216	0.390925	0.0	0.0	0.0	0.0	1.0
CityTier	4888.0	1.654255	0.916583	1.0	1.0	1.0	3.0	3.0
NumberOfPersonVisiting	4888.0	2.905074	0.724891	1.0	2.0	3.0	3.0	5.0
NumberOfFollowups	4843.0	3.708445	1.002509	1.0	3.0	4.0	4.0	6.0
PreferredPropertyStar	4862.0	3.581037	0.798009	3.0	3.0	3.0	4.0	5.0
NumberOfTrips	4748.0	3.236521	1.849019	1.0	2.0	3.0	4.0	22.0
Passport	4888.0	0.290917	0.454232	0.0	0.0	0.0	1.0	1.0
PitchSatisfactionScore	4888.0	3.078151	1.365792	1.0	2.0	3.0	4.0	5.0
OwnCar	4888.0	0.620295	0.485363	0.0	0.0	1.0	1.0	1.0
NumberOfChildrenVisiting	4822.0	1.187267	0.857861	0.0	1.0	1.0	2.0	3.0

Tabel Describe Kolom Categorical

	count	unique	top	freq
TypeofContact	4863	2	Self Enquiry	3444
Occupation	4888	4	Salaried	2368
Gender	4888	2	Male	2916
ProductPitched	4888	5	Basic	1842
MaritalStatus	4888	4	Married	2340
Designation	4888	5	Executive	1842

Tabel Describe Kolom Numerical

	count	mean	std	min	25%	50%	75%	max
Age	4662.0	37.622265	9.316387	18.0	31.0	36.0	44.0	61.0
DurationOfPitch	4637.0	15.490835	8.519643	5.0	9.0	13.0	20.0	127.0
MonthlyIncome	4655.0	23619.853491	5380.698361	1000.0	20346.0	22347.0	25571.0	98678.0

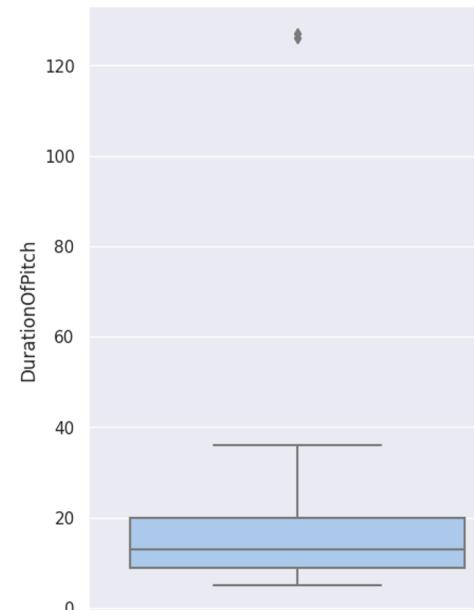
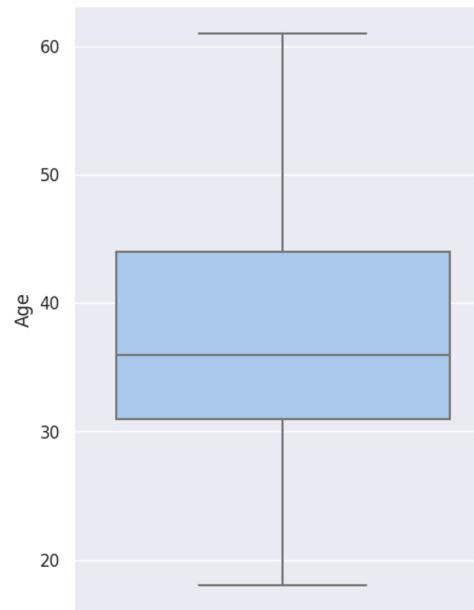
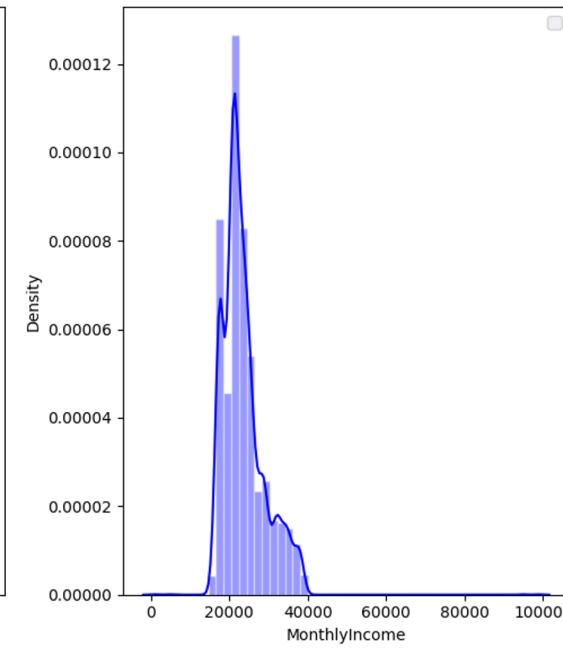
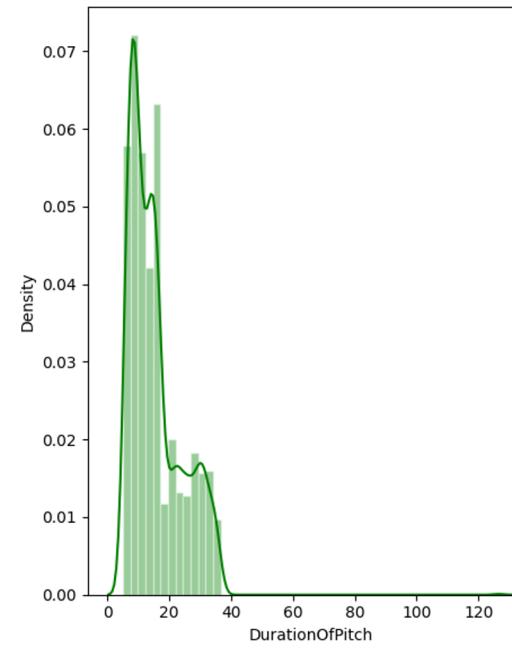
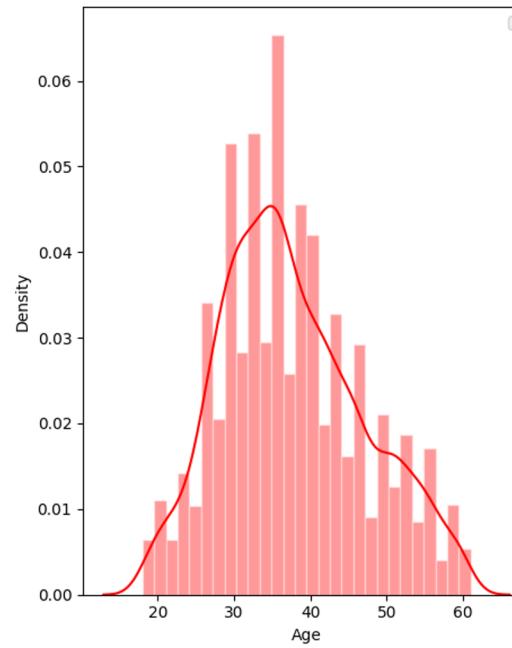
1C. Kolom yang memiliki nilai summary agak aneh

- Pada kolom *NumberOfTrips*, terdapat nilai max = 22 dan min = 1. Sedangkan mean (3.23) lebih besar dari median (3.0). Hal ini mengindikasikan data tersebar pada nilai-nilai yang memiliki value yang kecil, atau dapat disebut terdistribusi positive skewed.
- Kolom *MonthlyIncome* memiliki mean yang lebih besar dibandingkan nilai mediannya, yang mengindikasikan sebaran data tersebut yaitu positive skewed.
- Pada kolom *MonthlyIncome* pun terlihat memiliki standard deviation yang sangat tinggi, yang mengartikan terdapat data-data yang tersebar sangat jauh dari nilai mean, yang kemungkinan menandakan adanya outlier.



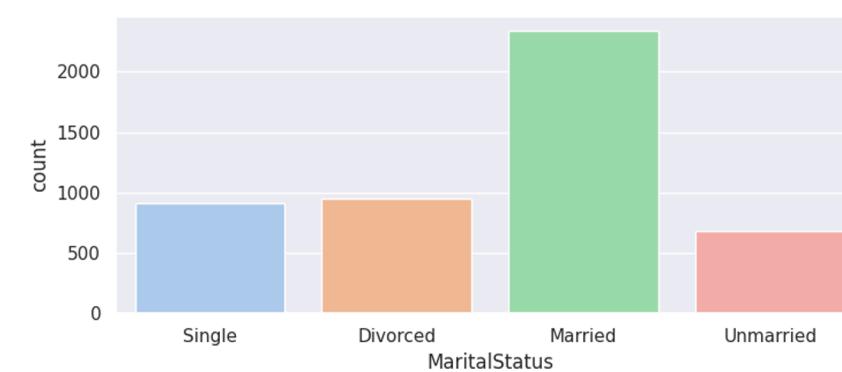
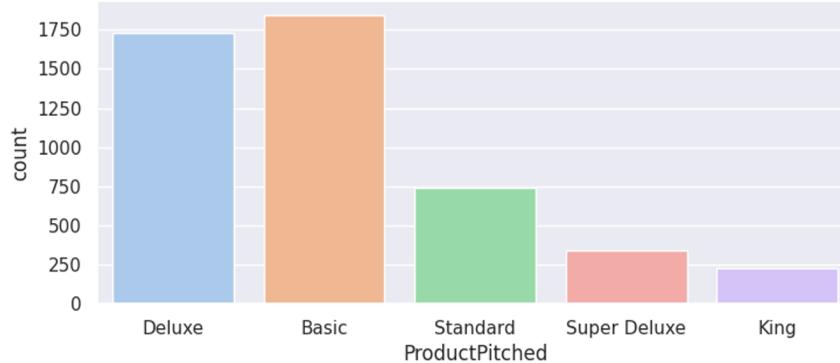
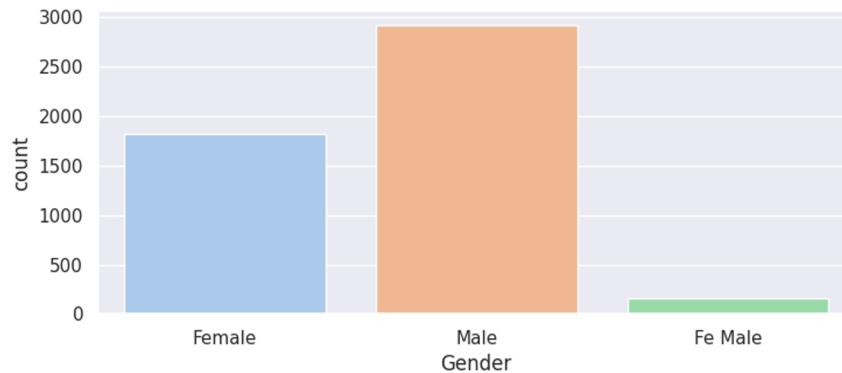
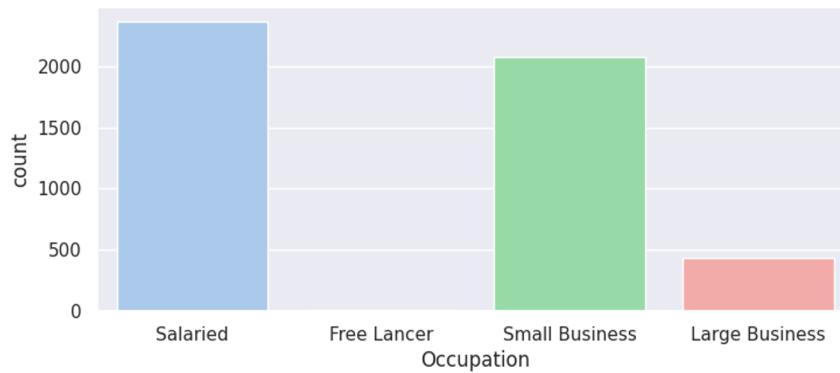
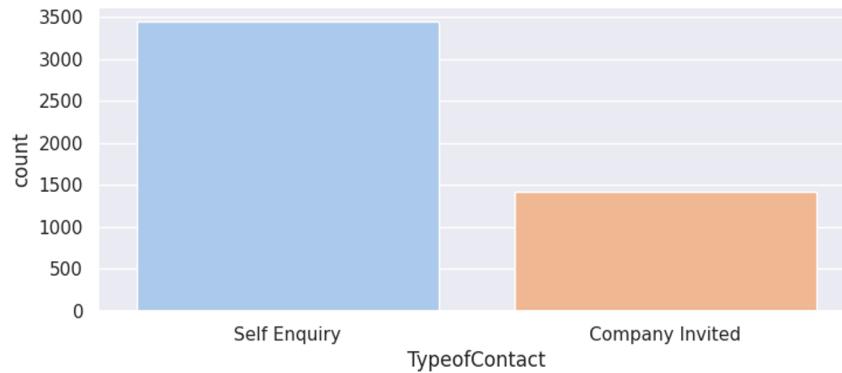
Univariate Analysis

Distribusi Kolom Numerical

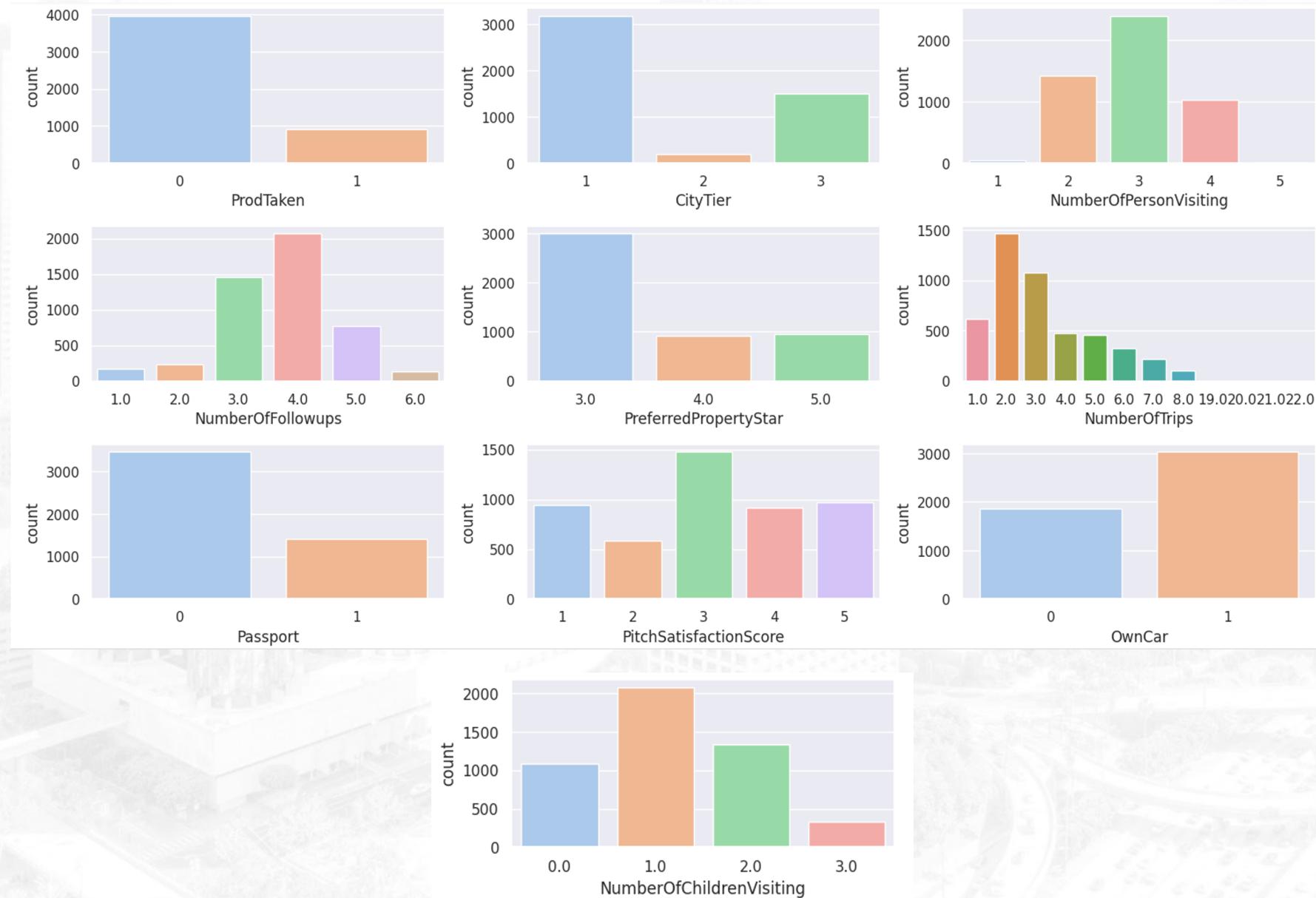


Boxplot Kolom Numerical

Countplot Kolom Categorical



Countplot Kolom Diskrit

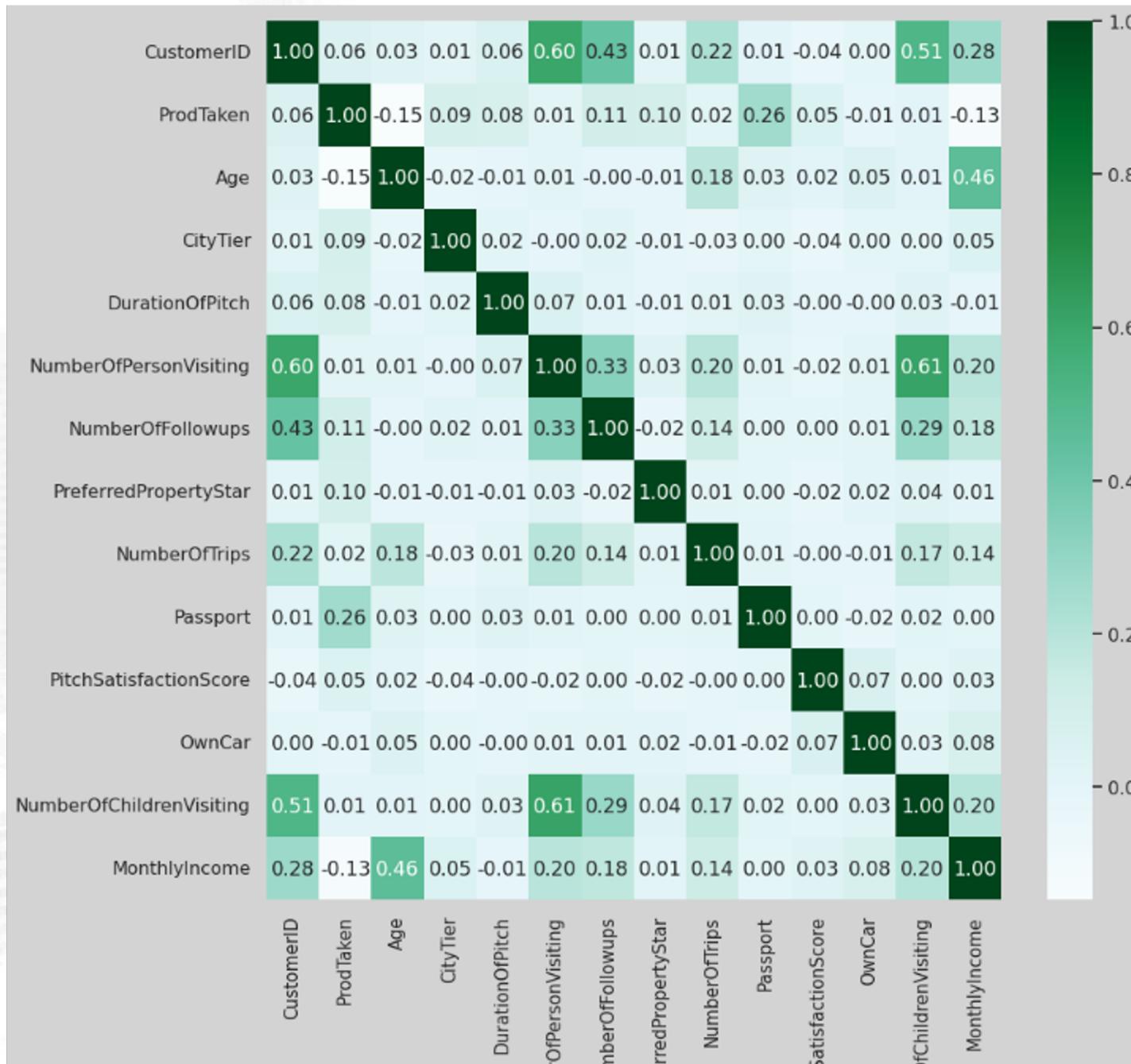


2. Hasil Observasi Univariate Analysis

- Pada kolom *Age* distribusi datanya hampir mendekati normal.
- Pada kolom *DurationOfPitch* terlihat distribusinya positive skewed, dimana kebanyakan frekuensi data tersebar pada nilai < 20. Terlihat juga adanya outlier pada nilai > 120, sehingga nantinya akan kita ubah distribusinya menjadi normal pada saat pre-processing data.
- Pada kolom *MonthlyIncome* data tersebar pada nilai 20k-30k dan terdapat data outlier pada nilai < 10k dan > 80k. Sama seperti kolom *DurationOfPitch*, nantinya akan kita ubah distribusi datanya menjadi normal pada saat melakukan pre-processing data.
- Terdapat outlier pada kolom *NumberOfTrips*, dimana adanya nilai ekstrim yang mencapai < 19, sehingga nanti akan kita hilangkan outlier-nya pada saat pre-processing.
- Customer yang memiliki *Occupation* sebagai freelancer sangat sedikit (0,04%) jika dibandingkan dengan *Occupation* yang lain.
- Pada kolom target *ProdTaken*, terdapat class imbalance atau ketimpangan data, dimana customer yang mengambil paket liburan (*ProdTaken* = 1) memiliki frekuensi di bawah 1000. Sedangkan customer yang tidak mengambil paket wisata (*ProdTaken* = 0) berjumlah sangat dominan, yaitu sekitar 4000 customer, sehingga ketimpangan data pada target harus diproses dengan undersampling atau oversampling pada saat data pre-processing nanti.



Multivariate Analysis



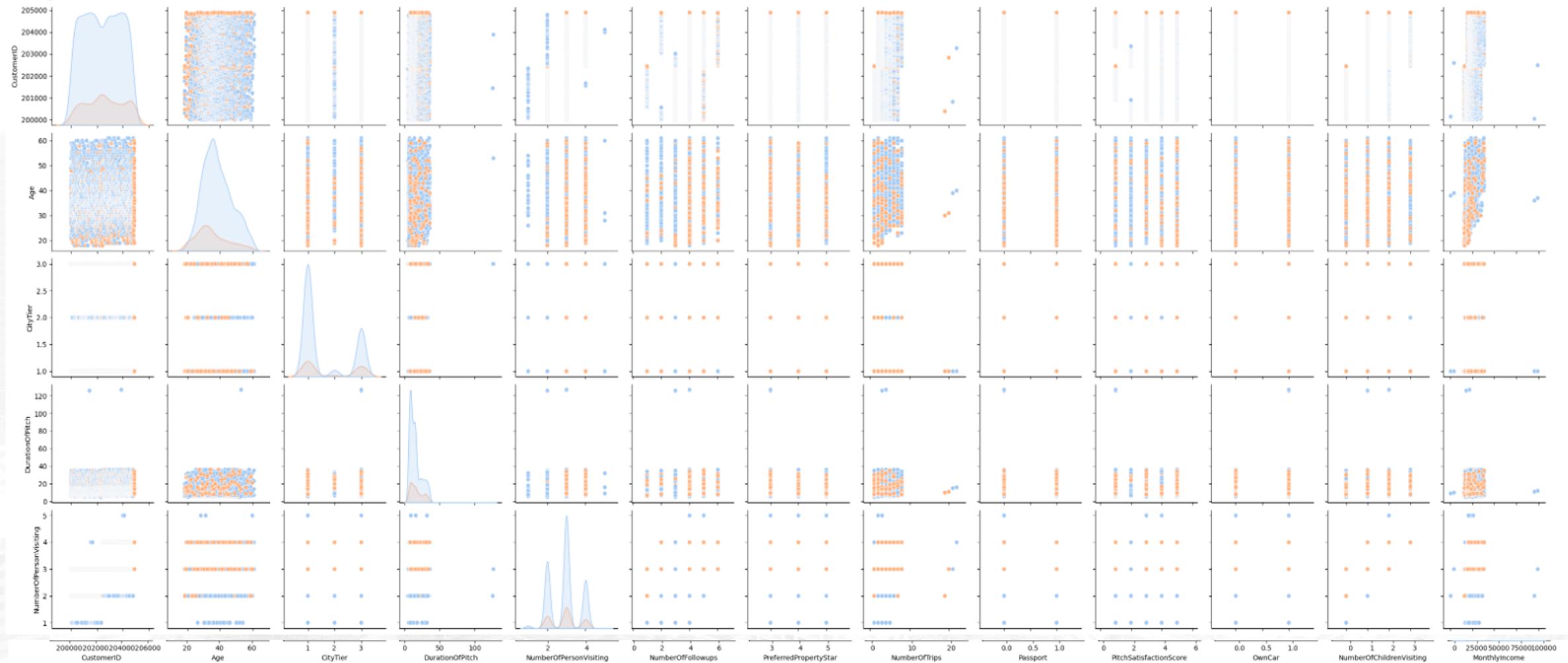
3. Hasil Observasi Multivariate Analysis

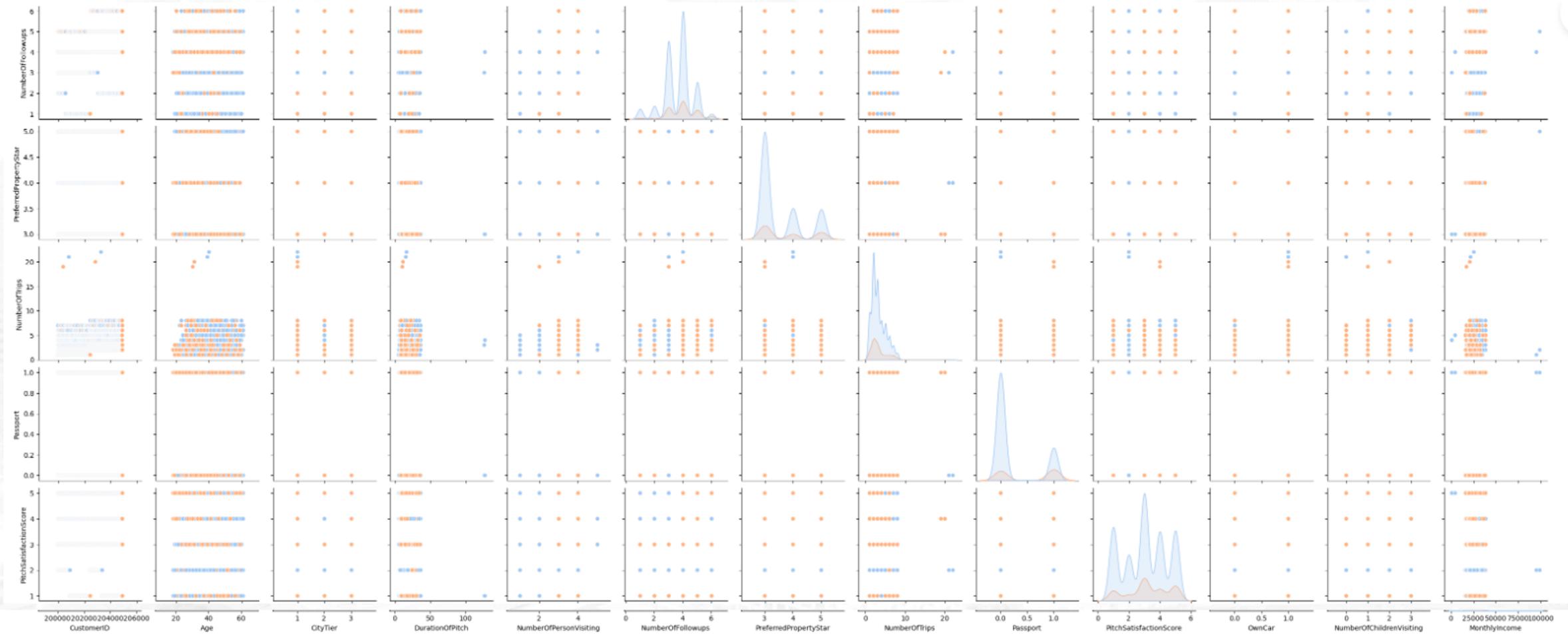
Correlation Heatmap

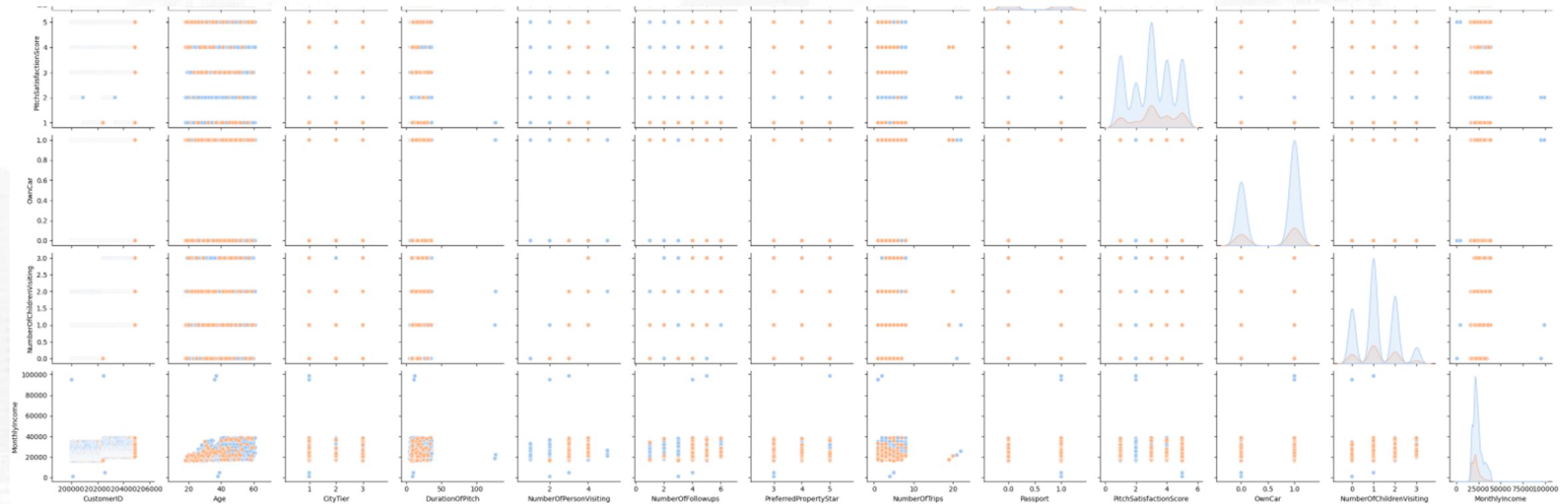
Dapat dilihat nilai korelasi positif (walaupun lemah) pada variable target kita yaitu *ProdTaken* dengan variabel *Passport*. Artinya semakin tinggi nilai passport customer (1), maka semakin tinggi pula nilai product taken-nya. Dengan kata lain, customer yang memiliki passport lebih cenderung membeli tawaran paket dibandingkan dengan customer yang tidak memiliki passport.

Grafik Pairplot

<Figure size 1000x1000 with 0 Axes>

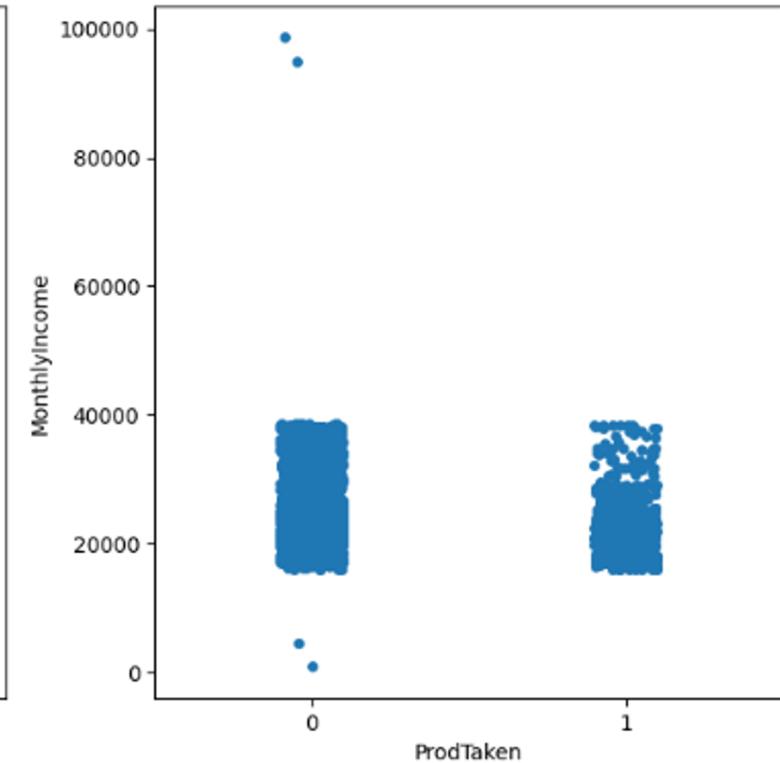
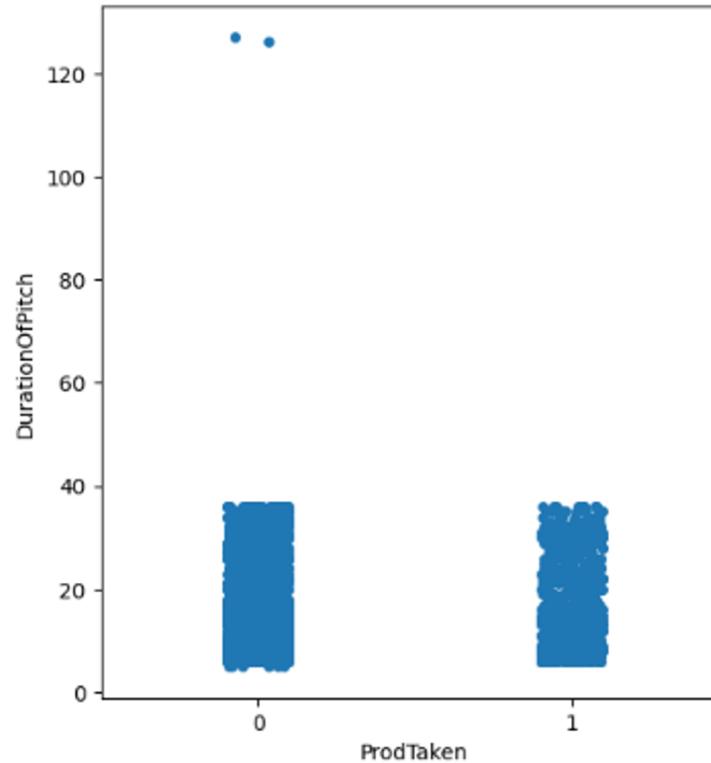
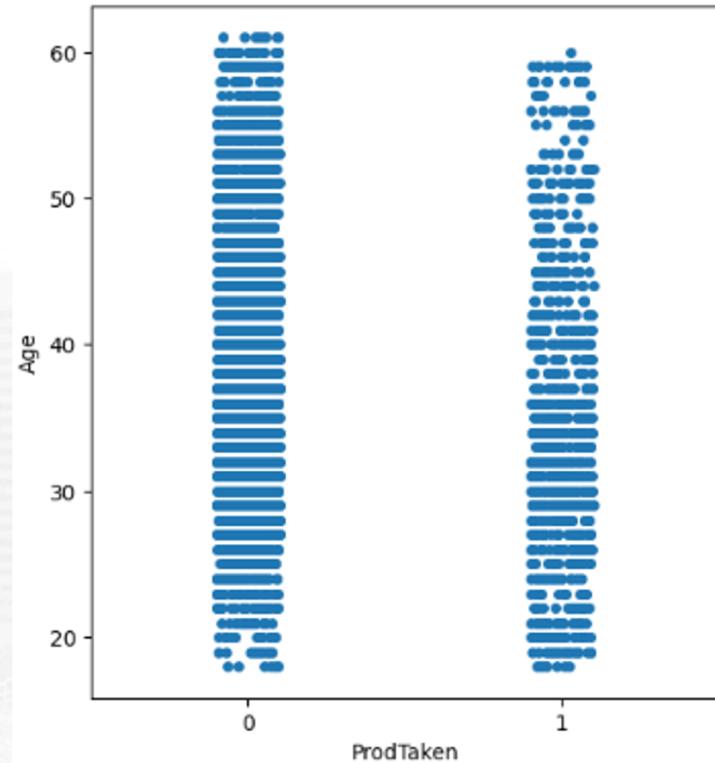






Dari grafik tersebut, terdapat 2 fitur yang memiliki korelasi yang cukup kuat, yaitu *NumberOfPersonVisiting* dan *NumberOfChildrenVisiting* yang membentuk korelasi positif. Pada proses feature selection, kita dapat mengeliminasi fitur yang redundan (mempunyai informasi serupa) agar tidak terjadi overfitting pada model machine learning nantinya.

Catplot/Stripplot Kolom Numerical

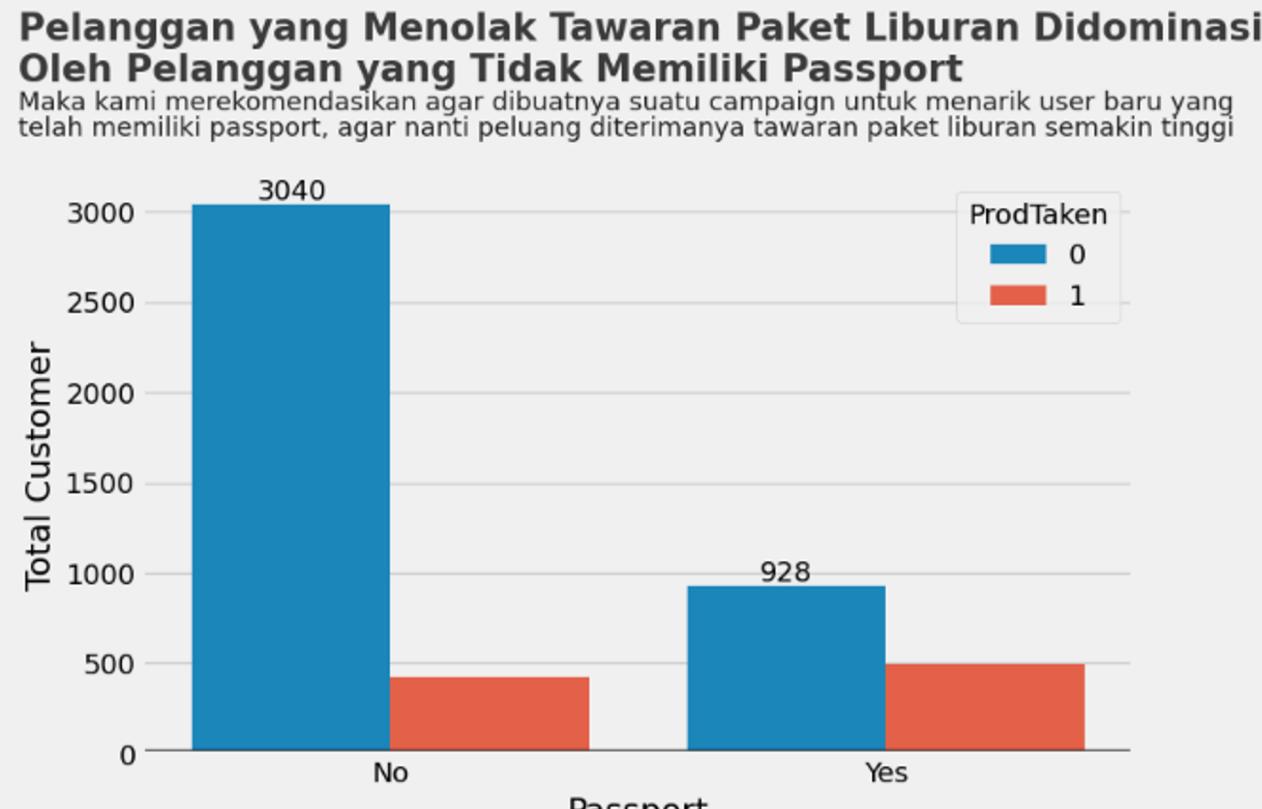


Terdapat korelasi negatif antara variabel *Age* terhadap *ProdTaken* dan variabel *MonthlyIncome* terhadap *ProdTaken*, yang berarti semakin kecil nilai *age* atau *monthly income*-nya, maka semakin besar nilai *product taken*-nya. Dengan kata lain, customer yang berumur kecil/muda lebih cenderung membeli tawaran paket liburan dibandingkan dengan customer yang berumur tua, dan customer yang bergaji kecil cenderung membeli tawaran paket liburan dibandingkan dengan customer yang bergaji besar.



Business Insight

4A. Grafik kepemilikan passport dan jumlah pelanggan yang menolak/membeli tawaran paket liburan



Dari grafik tersebut, dapat dilihat bahwa pelanggan yang menolak tawaran paket liburan didominasi oleh pelanggan yang tidak memiliki passport.

Rekomendasi bisnis yang kami sarankan adalah dibuatnya suatu campaign untuk menarik pelanggan baru yang telah memiliki passport, sehingga peluang diterimanya tawaran paket liburan semakin tinggi.

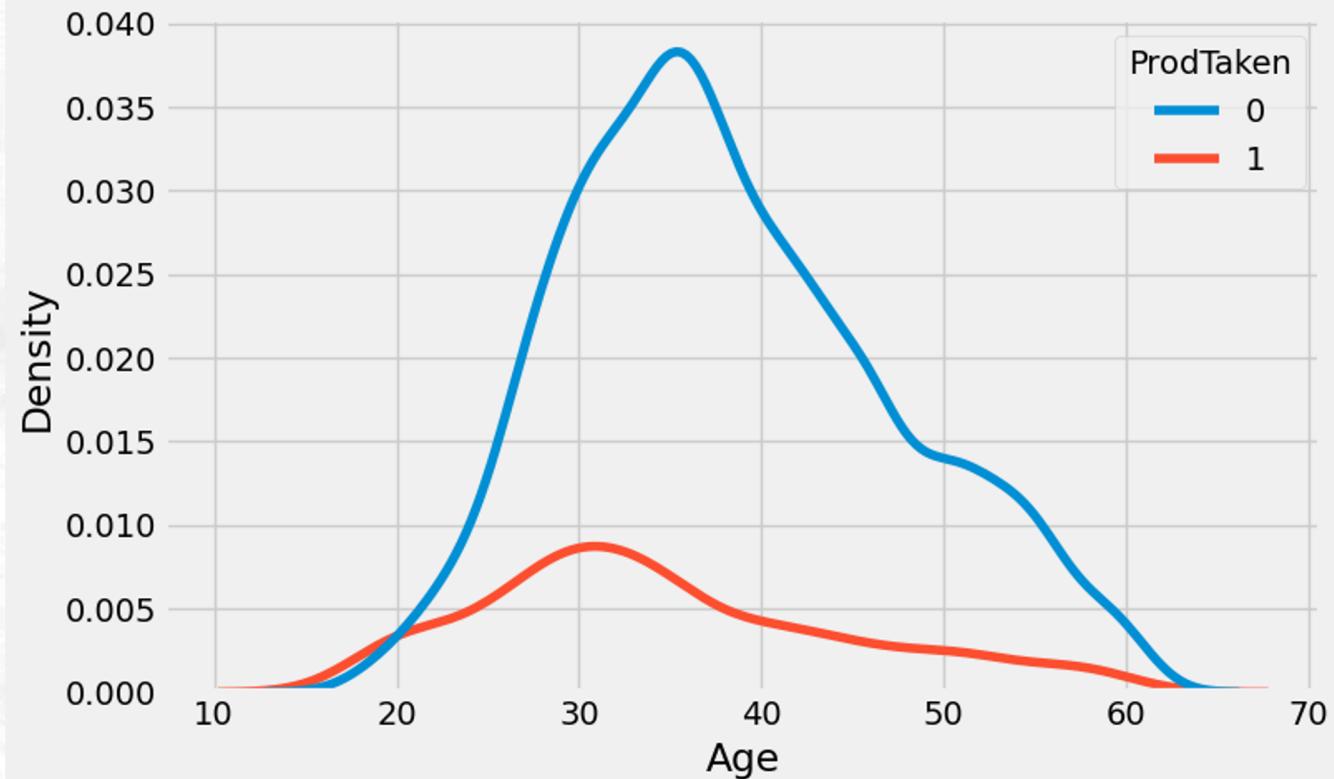
4B. Distribusi umur dan keputusan pelanggan untuk menolak/membeli tawaran paket liburan

Dari grafik tersebut, dapat dilihat bahwa pelanggan yang berumur muda (sekitar 20-35 tahun) yang paling banyak membeli tawaran paket liburan.

Rekomendasi bisnis yang kami sarankan adalah dibuatnya suatu campaign untuk menarik user baru yang berumur muda, agar nanti peluang diterimanya tawaran paket liburan semakin tinggi.

Pelanggan yang Berumur Muda ($\pm 20-35$ tahun) Cenderung Lebih Membeli Tawaran Paket Liburan

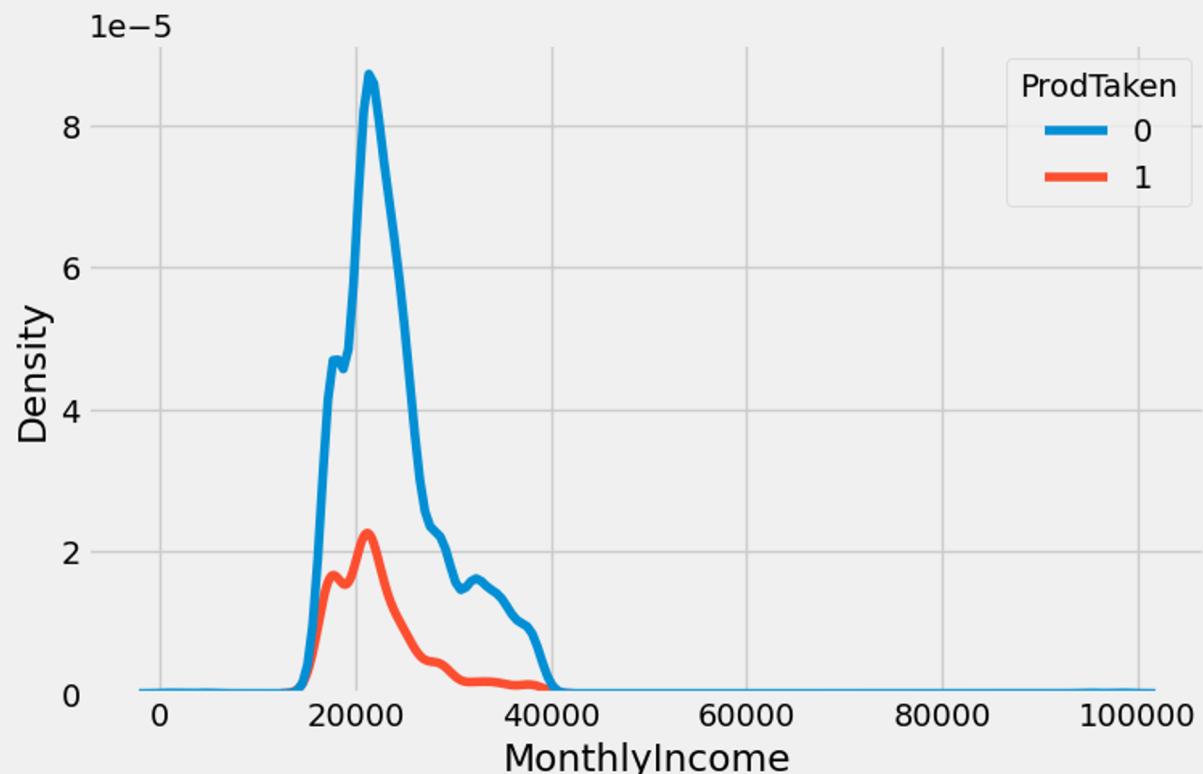
Maka kami merekomendasikan agar dibuatnya suatu campaign untuk menarik user baru yang berumur muda, agar nanti peluang diterimanya tawaran paket liburan semakin tinggi



4C. Distribusi gaji dan keputusan pelanggan untuk menolak/membeli tawaran paket liburan

Pelanggan yang Bergaji Tinggi / Besar Lebih Cenderung Untuk Menolak Tawaran Paket Liburan

Rekomendasi kami sama seperti sebelumnya, yaitu agar dibuatnya suatu campaign untuk menarik user baru yang berumur muda karena pelanggan yang berumur muda tidak memiliki gaji sebesar pelanggan yang berumur tua



Dari grafik tersebut, dapat dilihat bahwa pelanggan yang bergaji tinggi/besar lebih cenderung untuk menolak tawaran paket liburan.

Rekomendasi bisnis yang kami sarankan adalah sama seperti sebelumnya, yaitu dibuatnya suatu campaign untuk menarik user baru yang berumur muda karena pelanggan yang berumur muda tidak memiliki gaji sebesar pelanggan yang berumur tua.



Repository Git

Notebook Repository Git

Astryon / final_project Public

Watch 2 Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags Go to file Add file Code

Astryon Update README.md 6ab1444 11 hours ago 8 commits

Pandas_Lovers_Holiday_Package_Pre... Dibuat menggunakan Colaboratory 15 hours ago

README.md Update README.md 11 hours ago

README.md

Judul Project : HOLIDAY PACKAGE PREDICTION

Homework_EDA (Stage 1) Pandas_Lovers

About

No description, website, or topics provided.

Readme

0 stars

2 watching

0 forks

Report repository

Releases

No releases published

Berikut link repository git: https://github.com/Astryon/final_project

Stage 2

DATA PRE-PROCESSING



Data Cleansing

1A. Handling Missing Values



Before

CustomerID	0
ProdTaken	0
Age	226
TypeofContact	25
CityTier	0
DurationOfPitch	251
Occupation	0
Gender	0
NumberOfPersonVisiting	0
NumberOfFollowups	45
ProductPitched	0
PreferredPropertyStar	26
MaritalStatus	0
NumberOfTrips	140
Passport	0
PitchSatisfactionScore	0
OwnCar	0
NumberOfChildrenVisiting	66
Designation	0
MonthlyIncome	233
dtype:	int64

After

CustomerID	0
ProdTaken	0
Age	0
TypeofContact	0
CityTier	0
DurationOfPitch	0
Occupation	0
Gender	0
NumberOfPersonVisiting	0
NumberOfFollowups	0
ProductPitched	0
PreferredPropertyStar	0
MaritalStatus	0
NumberOfTrips	0
Passport	0
PitchSatisfactionScore	0
OwnCar	0
NumberOfChildrenVisiting	0
Designation	0
MonthlyIncome	0
dtype:	int64

Kolom yang terdapat nilai kosong (missing value) ada **8 kolom**.

Perlakuan pada masing-masing kolom yaitu:

1. Age : mean
2. TypeofContact : drop
3. DurationOfPitch : median
4. NumberOfFollowups : modus
5. PreferredPropertyStar : drop
6. NumberOfTrips : drop
7. NumberOfChildrenVisiting : drop
8. MonthlyIncome : median

Handling Invalid Data Type

Before

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4631 entries, 0 to 4887
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CustomerID      4631 non-null   int64  
 1   ProdTaken       4631 non-null   int64  
 2   Age              4631 non-null   float64 
 3   TypeofContact   4631 non-null   object  
 4   CityTier         4631 non-null   int64  
 5   DurationOfPitch 4631 non-null   float64 
 6   Occupation       4631 non-null   object  
 7   Gender            4631 non-null   object  
 8   NumberOfPersonVisiting 4631 non-null   int64  
 9   NumberOfFollowups 4631 non-null   float64 
 10  ProductPitched   4631 non-null   object  
 11  PreferredPropertyStar 4631 non-null   float64 
 12  MaritalStatus     4631 non-null   object  
 13  NumberOfTrips     4631 non-null   float64 
 14  Passport          4631 non-null   int64  
 15  PitchSatisfactionScore 4631 non-null   int64  
 16  OwnCar            4631 non-null   int64  
 17  NumberOfChildrenVisiting 4631 non-null   float64 
 18  Designation        4631 non-null   object  
 19  MonthlyIncome      4631 non-null   float64 

dtypes: float64(7), int64(7), object(6)
memory usage: 759.8+ KB
```

After

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4631 entries, 0 to 4887
Data columns (total 20 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   CustomerID      4631 non-null   int64  
 1   ProdTaken       4631 non-null   int64  
 2   Age              4631 non-null   int64  
 3   TypeofContact   4631 non-null   object  
 4   CityTier         4631 non-null   int64  
 5   DurationOfPitch 4631 non-null   int64  
 6   Occupation       4631 non-null   object  
 7   Gender            4631 non-null   object  
 8   NumberOfPersonVisiting 4631 non-null   int64  
 9   NumberOfFollowups 4631 non-null   int64  
 10  ProductPitched   4631 non-null   object  
 11  PreferredPropertyStar 4631 non-null   int64  
 12  MaritalStatus     4631 non-null   object  
 13  NumberOfTrips     4631 non-null   int64  
 14  Passport          4631 non-null   int64  
 15  PitchSatisfactionScore 4631 non-null   int64  
 16  OwnCar            4631 non-null   int64  
 17  NumberOfChildrenVisiting 4631 non-null   int64  
 18  Designation        4631 non-null   object  
 19  MonthlyIncome      4631 non-null   int64 

dtypes: int64(14), object(6)
memory usage: 759.8+ KB
```



Kolom yang tipe datanya diubah dari **float menjadi integer** sebanyak 7 kolom, yaitu *Age*, *DurationOfPitch*, *NumberOfFollowups*, *PreferredPropertyStar*, *NumberOfTrips*, *NumberOfChildrenVisiting*, dan *MonthlyIncome*.

1B. Handling Duplicated Data

```
print('Banyak data dan kolom : ',df_new.shape)
print('Jumlah ID Customer duplikat : ',df_new['CustomerID'].duplicated().sum())
print('Jumlah data duplikat : ',df_new.duplicated().sum())
```

Banyak data dan kolom : (4631, 20)
Jumlah ID Customer duplikat : 0
Jumlah data duplikat : 0

Tidak perlu ada yang di-handling karena tidak terdapat data duplikat pada dataset ini.

Handling Invalid Value

Pada kolom *Gender* terdapat kesalahan penulisan value '**Fe Male**'. Oleh karena itu diubah menjadi '**Female**'

```
# ubah value Fe Male menjadi Female
df_new = df_new.replace('Fe Male', 'Female')
```

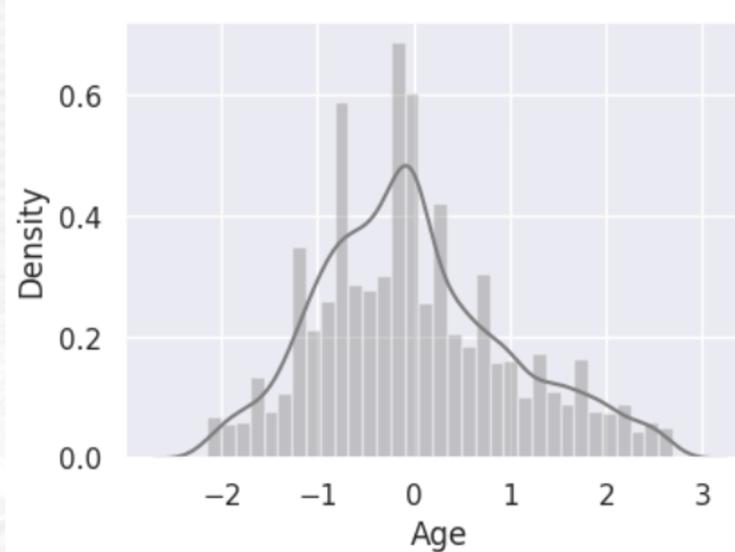
```
df_new['Gender'].value_counts()
```

Male	2765
Female	1866
Name:	Gender, dtype: int64

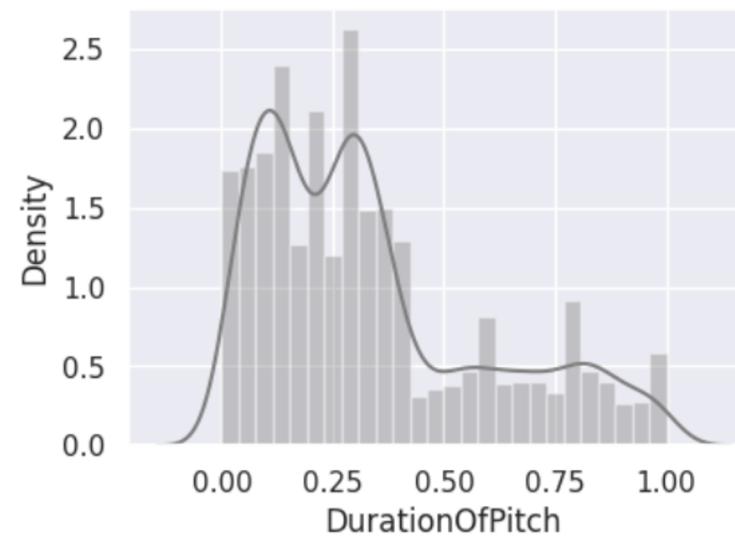
1C. Handling Outliers

Karena kami akan menggunakan **model non-linear** (model yang robust terhadap outliers), maka kami tidak akan menghilangkan data outliers-nya.

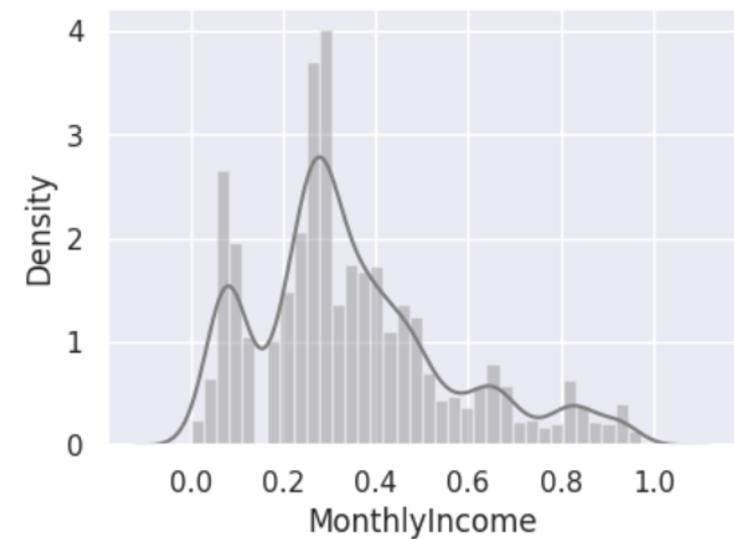
1D. Feature Transformation



Kolom Age:
standarisasi



Kolom DurationOfPitch:
normalisasi



Kolom MonthlyIncome:
normalisasi

1E. Feature Encoding

```
# onehots encoder pada Occupation dan TypeofContact
for cat in ['Occupation', 'TypeofContact']:
    onehots = pd.get_dummies(df_new[cat], prefix=cat)
    df_new = df_new.join(onehots)
```

One-hot encoding pada kolom *Occupation*
dan *TypeOfContact*

```
# label encoder pada Gender
df_new['Gender'] = df_new['Gender'].astype('category').cat.codes
```

```
# membuat function Product
def product(x):
    if x['ProductPitched'] == 'Basic':
        product = 0
    if x['ProductPitched'] == 'Standard':
        product = 1
    if x['ProductPitched'] == 'Deluxe':
        product = 2
    if x['ProductPitched'] == 'Super Deluxe':
        product = 3
    if x['ProductPitched'] == 'King':
        product = 4
    return product
```

```
# menjalankan function Product
df_new['ProductPitched'] = df_new.apply(lambda x: product(x), axis=1)
```

```
# membuat function MaritalStatus
def status(x):
    if x['MaritalStatus'] == 'Married':
        status = 1
    else:
        status = 0
    return status

# membuat kolom baru bernama 'IsMarried' yang menjalankan function grouping
df_new['IsMarried'] = df_new.apply(lambda x: status(x), axis=1)
```

Label encoding pada
kolom *Gender*,
ProductPitched,
MaritalStatus, dan
Designation

```
# membuat function Designation
def designation(x):
    if x['Designation'] == 'Executive':
        designation = 0
    if x['Designation'] == 'Manager':
        designation = 1
    if x['Designation'] == 'Senior Manager':
        designation = 2
    if x['Designation'] == 'AVP':
        designation = 3
    if x['Designation'] == 'VP':
        designation = 4
    return designation
```

```
df_new['Designation'] = df_new.apply(lambda x: designation(x), axis=1)
df_new.tail()
```

2A. Feature Selection

Feature selection pertama

Awalnya, kami memilih fitur berdasarkan nilai korelasi terhadap *ProdTaken*. Fitur dengan korelasi $> 0,1$ dipilih dan dilanjutkan untuk proses modeling. Namun, ternyata hasil modeling dan evaluation nilai precision-nya tidak bagus, sehingga kami melakukan iterasi kembali ke proses feature selection.

Feature selection kedua

1. Drop fitur yg tidak relevan

```
# drop feature yang tidak terpakai
features = df_new.drop(columns=['CustomerID', 'Occupation', 'TypeofContact', 'MaritalStatus', 'Occupation_Salaried',
                                'Occupation_Large Business', 'Occupation_Free Lancer', 'Occupation_Small Business',
                                'TypeofContact_Company Invited', 'TypeofContact_Self Enquiry', 'OwnCar']).copy()
```

- Fitur *CustomerID* didrop karena fitur tidak dipakai untuk modeling.
- Fitur *Occupation*, *TypeofContact* & *MaritalStatus* didrop karena sudah di-encode, sehingga fitur yang aslinya tidak terpakai.
- Fitur *Occupation_Salaried*, *Occupation_Large Business*, *Occupation_Free Lancer*, *Occupation_Small Business*, *TypeofContact_Company Invited*, *TypeofContact_Self Enquiry* & *OwnCar* didrop karena korelasi fiturnya kecil terhadap kolom target dan juga redundant.

2. Train semua fitur menggunakan algoritma XGBoost (algoritma ini merupakan yang terbaik pada feature selection pertama)

```
[ ] #cek precision fitur menggunakan xgboost
from xgboost import XGBClassifier

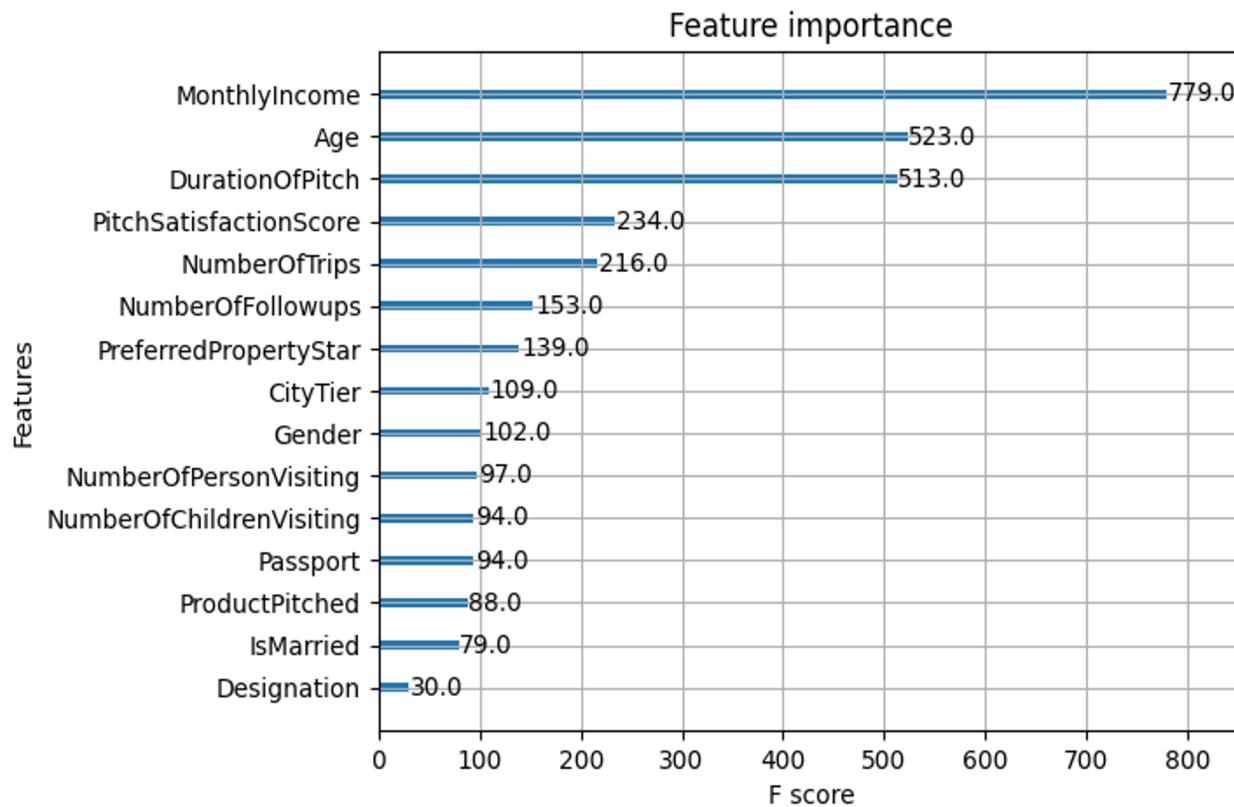
xg = XGBClassifier(random_state=42)
xg.fit(X_train, y_train)
eval_classification(xg)

F1-Score (Test Set): 0.74
roc_auc (test-proba): 0.92
Recall (Test Set): 0.67
Accuracy (Test Set): 0.92

Precision (Test Set): 0.83
Precision (Train Set): 1.00
```

3. Drop kembali fitur yang kurang relevan

Walaupun hasil dari evaluation sudah cukup baik, tetapi modelnya masih overfitting. Oleh karena itu, kami akan drop lagi beberapa fitur yang kurang relevan dengan melihat dari feature importance-nya.



```
[ ] # drop fitur yg tidak terlalu penting
features = features.drop(columns=[ 'Designation', 'NumberOfChildrenVisiting', 'IsMarried',
                                    'NumberOfPersonVisiting', 'Gender' ]).copy()
```

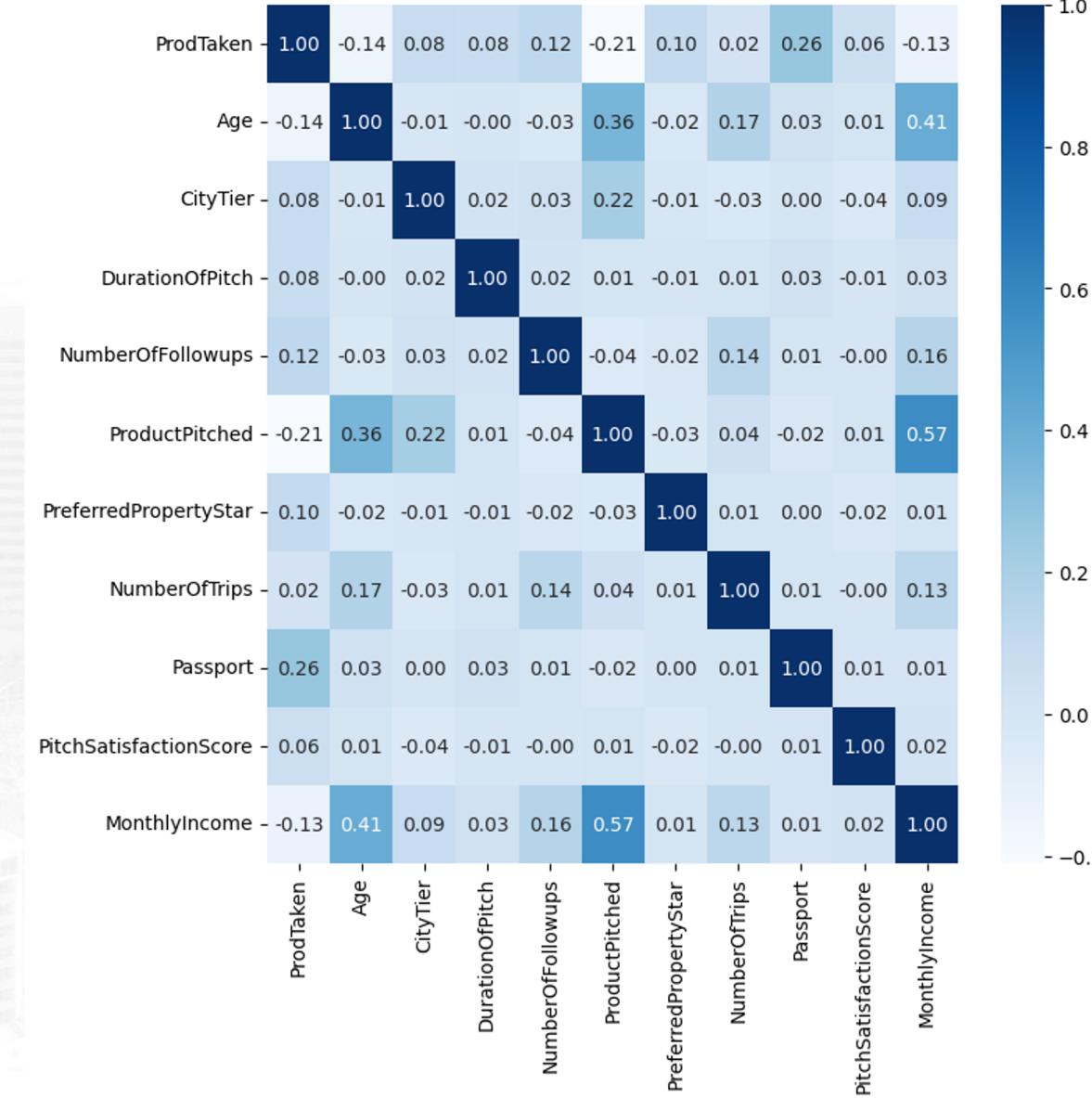
Kami memfokuskan pada 7 fitur terbawah. Kami tetap mengambil fitur *Passport* dan *ProductPitched*. Lima fitur sisanya tidak dipakai.

- Fitur *Passport* dipertahankan karena melihat business insight pada EDA.
- Fitur *ProductPitched* dipertahankan karena fitur ini dapat menurunkan nilai precision jika didrop.

```
[ ] # cek fitur
features.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 4631 entries, 0 to 4887
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   ProdTaken        4631 non-null   int64  
 1   Age              4631 non-null   float64 
 2   CityTier         4631 non-null   int64  
 3   DurationOfPitch 4631 non-null   float64 
 4   NumberOfFollowups 4631 non-null   int64  
 5   ProductPitched   4631 non-null   int64  
 6   PreferredPropertyStar 4631 non-null   int64  
 7   NumberOfTrips    4631 non-null   int64  
 8   Passport          4631 non-null   int64  
 9   PitchSatisfactionScore 4631 non-null   int64  
 10  MonthlyIncome     4631 non-null   float64 
dtypes: float64(3), int64(8)
memory usage: 563.2 KB
```

Hasil akhir
feature selection



4. Cek korelasi fitur

Dapat dilihat pada heatmap di samping bahwa **tidak terdapat fitur yang redundant**, sehingga tidak ada fitur yang harus didrop.

2B. Feature Extraction

```
# membuat function MaritalStatus
def status(x):
    if x['MaritalStatus'] == 'Married':
        status = 1
    else:
        status = 0
    return status

# membuat kolom baru bernama 'IsMarried' yang menjalankan function grouping
df_new['IsMarried'] = df_new.apply(lambda x: status(x), axis=1)
```

Membuat fitur baru *IsMarried* yang diambil dari fitur *MaritalStatus* dengan ketentuan:

- **value 0** untuk ‘Single’, ‘Unmarried’, ‘Divorced’
- **value 1** untuk ‘Married’

2C. Feature Tambahan

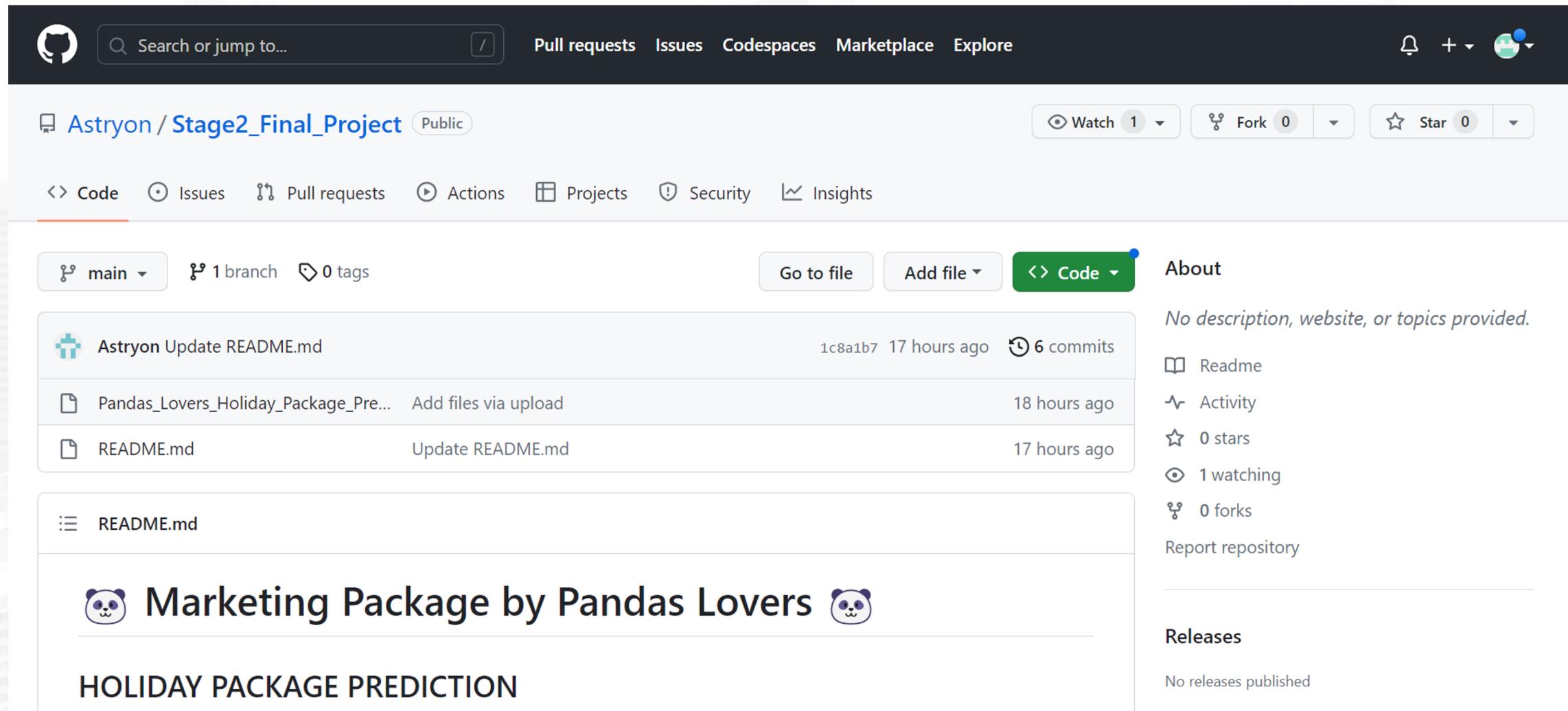
Berikut beberapa fitur tambahan yang mungkin akan dapat membantu meningkatkan performansi model.

- Harga Paket Liburan
- Metode Pembayaran
- Nomor atau Provider Telepon Customer
- Domisili atau Kota
- Durasi Paket Liburan



Repository Git

Notebook Repository Git



The screenshot shows a GitHub repository page. At the top, there's a navigation bar with a GitHub icon, a search bar containing "Search or jump to...", and links for Pull requests, Issues, Codespaces, Marketplace, and Explore. To the right of the search bar are icons for notifications, a plus sign, and a profile picture. Below the navigation bar, the repository name "Astryon / Stage2_Final_Project" is displayed, along with a "Public" badge, a "Watch" button (1 watch), a "Fork" button (0 forks), and a "Star" button (0 stars). A horizontal menu below the repository name includes links for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. The "Code" link is underlined, indicating it is the active tab. On the left side, there's a sidebar showing the main branch ("main"), 1 branch, and 0 tags. It also lists recent commits: "Astryon Update README.md" (1c8a1b7, 17 hours ago, 6 commits), "Pandas_Lovers_Holiday_Package_Pre..." (Add files via upload, 18 hours ago), and "README.md" (Update README.md, 17 hours ago). Below this, there's a section for "README.md" with a file icon. The main content area features a title "Marketing Package by Pandas Lovers" with two panda face icons on either side, followed by the subtitle "HOLIDAY PACKAGE PREDICTION". To the right of the main content, there's a sidebar titled "About" which states "No description, website, or topics provided." It also lists repository statistics: Readme, Activity, 0 stars, 1 watching, 0 forks, and a "Report repository" link. At the bottom of the sidebar, there's a "Releases" section stating "No releases published".

Berikut link repository git: https://github.com/Astryon/Stage2_Final_Project/

Stage 3

MACHINE LEARNING

MODELING & EVALUATION



Modeling

1A. Split Data Train & Test

```
# membagi data train dan test (70:30)
trainset, testset = train_test_split(features, test_size=0.3, random_state=42)
print('Jumlah baris train set :', len(trainset))
print('Jumlah baris test set :', len(testset))

Jumlah baris train set : 3241
Jumlah baris test set : 1390
```

1. Split data dengan rasio 70:30

```
# split x dan y
x_train = trainset.drop(['ProdTaken'],axis=1)
y_train = trainset['ProdTaken']
x_test = testset.drop(['ProdTaken'],axis=1)
y_test = testset['ProdTaken']
```

2. Split fitur dan target pada masing-masing data train dan data test

```
print('Jumlah data train sebelum SMOTE terdiri dari', len(x_train), 'baris')
print(f'Jumlah class 0 : {sum(y_train==0)}')
print(f'Jumlah class 1 : {sum(y_train==1)}\n')

# oversampling menggunakan SMOTE
smote = SMOTE(sampling_strategy=0.5,random_state=42)
x_train,y_train = smote.fit_resample(x_train,y_train)

print(f'Jumlah data train setelah SMOTE terdiri dari {len(x_train)} baris')
print(f'Jumlah class 0 : {sum(y_train==0)}')
print(f'Jumlah class 1 : {sum(y_train==1)}\n')

Jumlah data train sebelum SMOTE terdiri dari 3241 baris
Jumlah class 0 : 2596
Jumlah class 1 : 645

Jumlah data train setelah SMOTE terdiri dari 3894 baris
Jumlah class 0 : 2596
Jumlah class 1 : 1298
```

```
X = features.drop(columns=['ProdTaken']).copy()
y = features['ProdTaken']
X_train = x_train
X_test = x_test

print('Jumlah baris x_train :', len(x_train))
print('Jumlah baris y_train :', len(y_train))
print('Jumlah baris x_test :', len(x_test))
print('Jumlah baris y_test :', len(y_test))
```

```
Jumlah baris x_train : 3894
Jumlah baris y_train : 3894
Jumlah baris x_test : 1390
Jumlah baris y_test : 1390
```

3. Handling class imbalance pada data train dengan SMOTE (algoritma oversampling)

1B. Modeling

Metode

Kami mengolah dataset ini dengan metode **klasifikasi**. Kami ingin mendeteksi dan membedakan mana saja yang merupakan *potential customer* dan yang bukan. *Potential customer* ini nantinya akan diberikan tawaran paket liburan terbaru, sehingga diharapkan nantinya dapat meningkatkan *revenue* perusahaan dan mengoptimalkan *marketing cost*.

Metric

Metric yang digunakan adalah **precision**, karena kami ingin menurunkan angka false positive (FP). False positive adalah kesalahan memprediksi jumlah *potential customer*. Beberapa pelanggan diberikan penawaran paket liburan oleh perusahaan, namun pelanggan tersebut pada akhirnya tidak membeli paket tersebut. Hal ini tentu tidak efisien.

Algoritma

Algoritma yang diimplementasikan adalah **KNN, decision tree, random forest, AdaBoost, dan XGBoost**. Pemilihan algoritma ini dikarenakan model datanya yang non-linear. Dari kelima algoritma ini kemudian akan diseleksi hingga mendapatkan satu algoritma model yang terbaik.

1. KNN

```
# knn
from sklearn.neighbors import KNeighborsClassifier # import knn dari sklearn
knn = KNeighborsClassifier() # inisiasi object dengan nama knn
knn.fit(X_train, y_train) # fit model KNN dari data train
eval_classification(knn)

F1-Score (Test Set): 0.56
roc_auc (test-proba): 0.83
Recall (Test Set): 0.60
Accuracy (Test Set): 0.83

Precision (Test Set): 0.52
Precision (Train Set): 0.82
```

```
# decision tree
from sklearn.tree import DecisionTreeClassifier # import decision tree dari sklearn
dt = DecisionTreeClassifier() # inisiasi object dengan nama dt
dt.fit(X_train, y_train) # fit model decision tree dari data train
eval_classification(dt)
```

```
F1-Score (Test Set): 0.70
roc_auc (test-proba): 0.83
Recall (Test Set): 0.72
Accuracy (Test Set): 0.89

Precision (Test Set): 0.69
Precision (Train Set): 1.00
```

2. Decision Tree

```
from sklearn.ensemble import RandomForestClassifier  
  
rf = RandomForestClassifier()  
rf.fit(X_train, y_train)  
eval_classification(rf)
```

3. Random Forest

F1-Score (Test Set): 0.79
roc_auc (test-proba): 0.94
Recall (Test Set): 0.73
Accuracy (Test Set): 0.93

Precision (Test Set): 0.85
Precision (Train Set): 1.00

```
from sklearn.ensemble import AdaBoostClassifier  
  
clf = AdaBoostClassifier()  
clf.fit(X_train, y_train)  
eval_classification(clf)
```

F1-Score (Test Set): 0.48
roc_auc (test-proba): 0.77
Recall (Test Set): 0.47
Accuracy (Test Set): 0.82

Precision (Test Set): 0.49
Precision (Train Set): 0.70

4. AdaBoost

```
from xgboost import XGBClassifier  
  
xg = XGBClassifier(random_state=42)  
xg.fit(X_train, y_train)  
eval_classification(xg)
```

F1-Score (Test Set): 0.74
roc_auc (test-proba): 0.92
Recall (Test Set): 0.68
Accuracy (Test Set): 0.91

Precision (Test Set): 0.80
Precision (Train Set): 1.00

5. XGBoost

1C. Model Evaluation: Pemilihan dan Perhitungan Metrics Model

	Model	Precision (Test)	Precision (Train)
0	KNN	0.5211	0.8226
1	Decision Tree	0.7176	1.0000
2	Random Forest	0.8626	1.0000
3	AdaBoost	0.4916	0.6978
4	XGBoost	0.8048	1.0000

Karena hasil precision yang paling baik adalah model dari **random forest** dan **XGBoost**, maka kami pilih kedua model tersebut untuk dilanjutkan ke tahap hyperparameter tuning agar mendapatkan hasil precision yang lebih optimal.

1D. Model Evaluation: Validasi dengan Cross-validation

Random Forest

```
# cross validation
from sklearn.model_selection import cross_validate

model = rf
score_rf = cross_validate(model, X, y, cv=5, scoring='precision', return_train_score=True)
print('precision (train): ' + str(score_rf['train_score'].mean()))
print('precision (test): ' + str(score_rf['test_score'].mean()))

precision (train): 1.0
precision (test): 0.9522130549880774
```

XGBoost

```
# cross validation
from sklearn.model_selection import cross_validate
score_xg = cross_validate(xg, X, y, cv=5, scoring='precision', return_train_score=True)
print('precision (train): ' + str(score_xg['train_score'].mean()))
print('precision (test): ' + str(score_xg['test_score'].mean()))

precision (train): 1.0
precision (test): 0.915336193813275
```

Algoritma model menunjukkan indikasi **overfitting** karena skor data train terlalu tinggi, yaitu sama dengan 1. Overfitting terjadi ketika model terlalu cocok dengan data train, sehingga nantinya tidak dapat memprediksi data test yang tidak dikenal dengan akurat. Oleh karena itu, perlu dilakukan hyperparameter tuning.

1E. Hyperparameter Tuning

Random Forest

```
#setelah dilakukan hyperparameter tuning
rf_tuned = RandomForestClassifier(n_estimators=10,max_depth=28,criterion='entropy',random_state=42)
rf_tuned.fit(X_train, y_train)
eval_classification(rf_tuned)
```

XGBoost

```
#setelah dilakukan hyperparameter tuning
from xgboost import XGBClassifier
xg_tuned = XGBClassifier(n_estimators=25,max_depth=19,gamma=1,learning_rate=0.3693877551020408,random_state=42)
xg_tuned.fit(X_train, y_train)
eval_classification(xg_tuned)
```

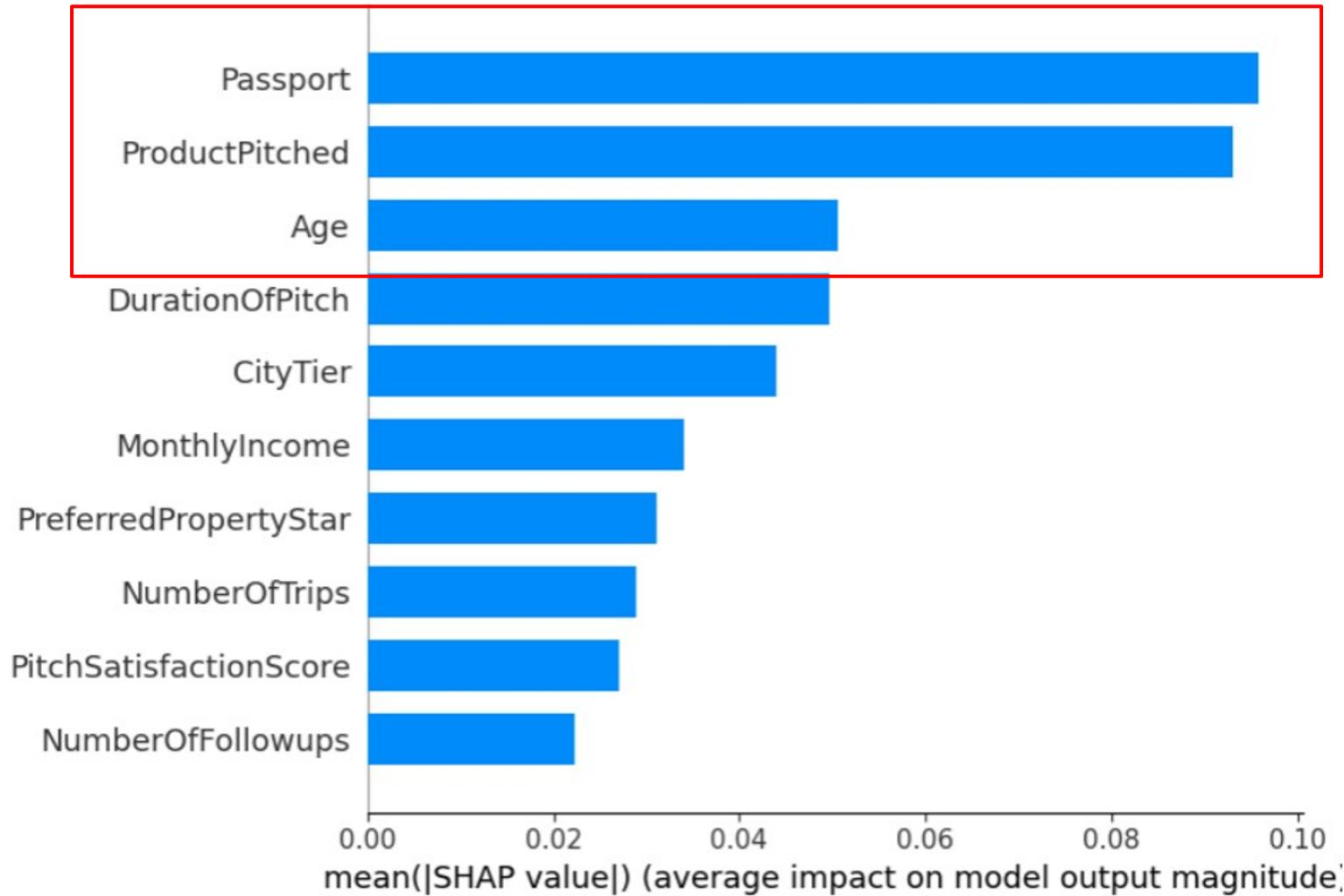
	Model	Precision (Test)	Precision (Train)
0	Random Forest	0.9405	0.9985
1	XGBoost	0.9152	0.9994

Pada hasil evaluasi di samping, model yang memiliki hasil paling baik adalah **random forest**, karena selisih nilai data test dan data train sangat kecil dan tidak lebih dari 0,1. Oleh karena itu, kami memilih random forest sebagai **best-fit model**.



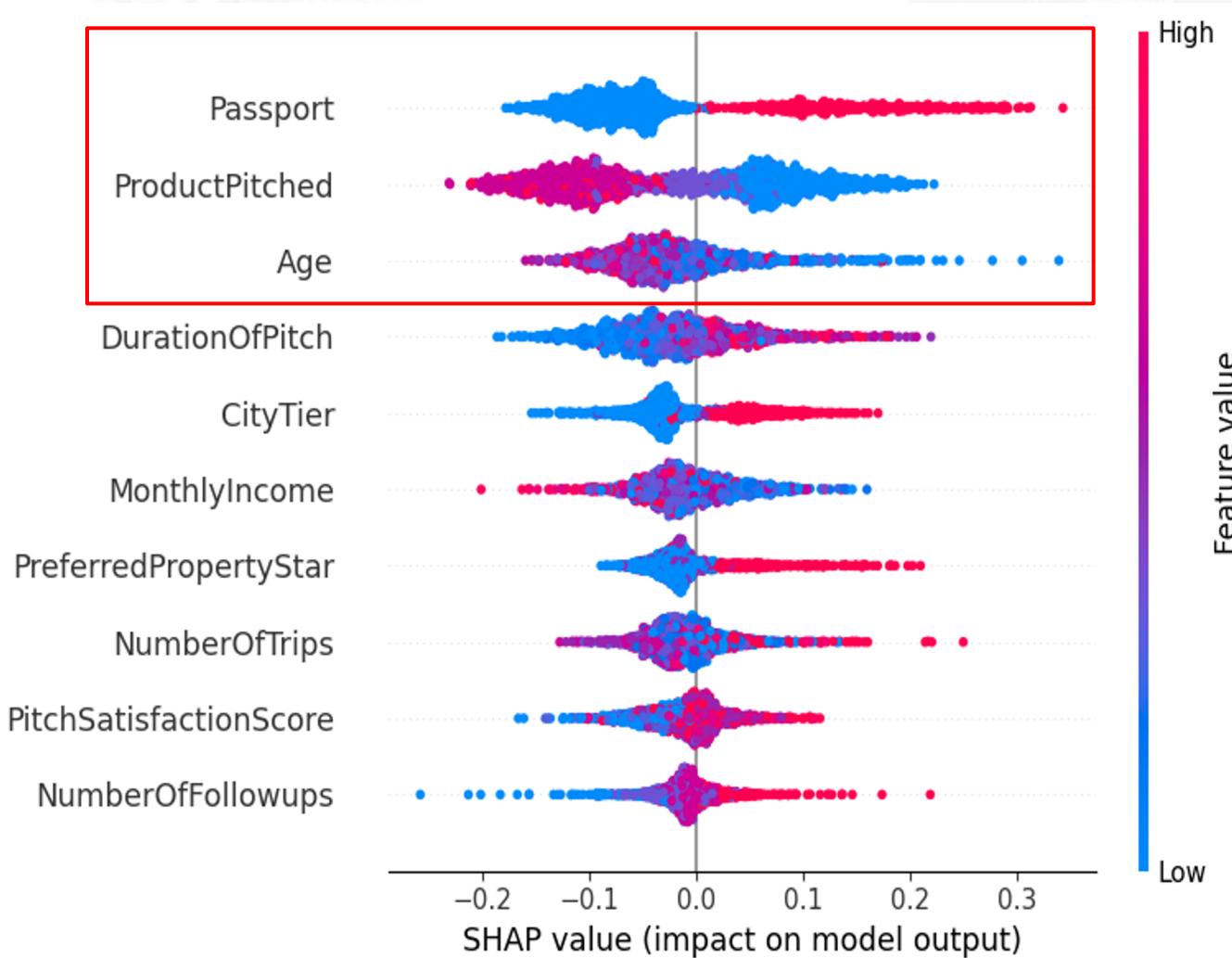
Feature Importance

Evaluasi Feature



Berikut adalah feature importance dari hasil modeling random forest yang sudah dilakukan hyperparameter tuning. Tiga fitur teratas yang paling penting yaitu **Passport**, **ProductPitched**, dan **Age**.

Business Insight

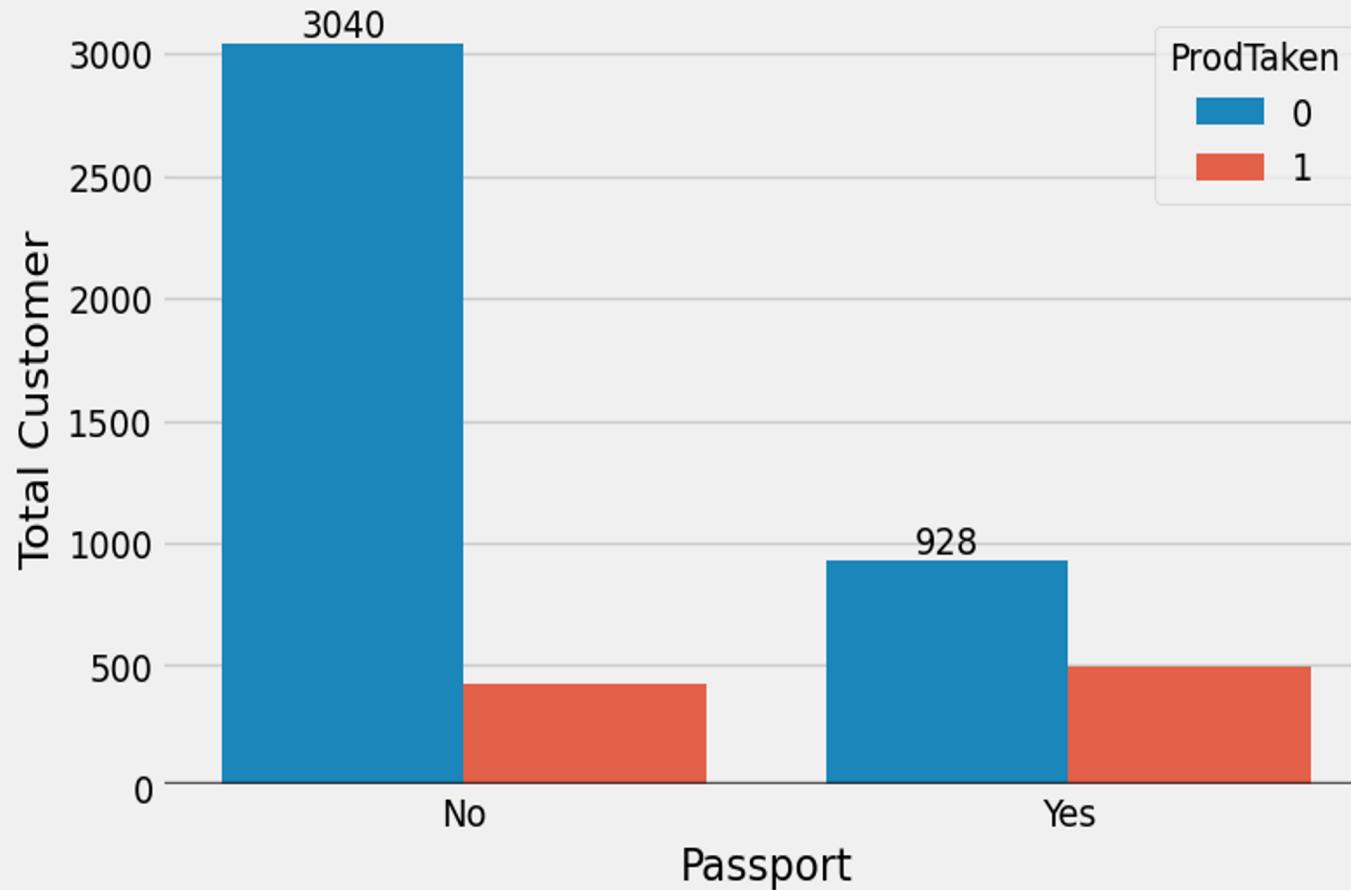


- **Passport:** Fitur *Passport*, menunjukkan pengaruh positif terhadap *ProdTaken*. Jadi, pelanggan yang memiliki passport cenderung untuk membeli paket liburan.
- **ProductPitched:** Fitur *ProductPitched* menunjukkan pengaruh negatif terhadap *ProdTaken*. Jadi, semakin kecil kelas paketnya, jumlah pembeliannya semakin besar.
- **Age:** Fitur *Age* menunjukkan pengaruh negatif terhadap *ProdTaken*. Jadi, pelanggan dengan rentang usia muda ($\pm 20\text{-}35$ tahun) lebih cenderung untuk membeli paket.

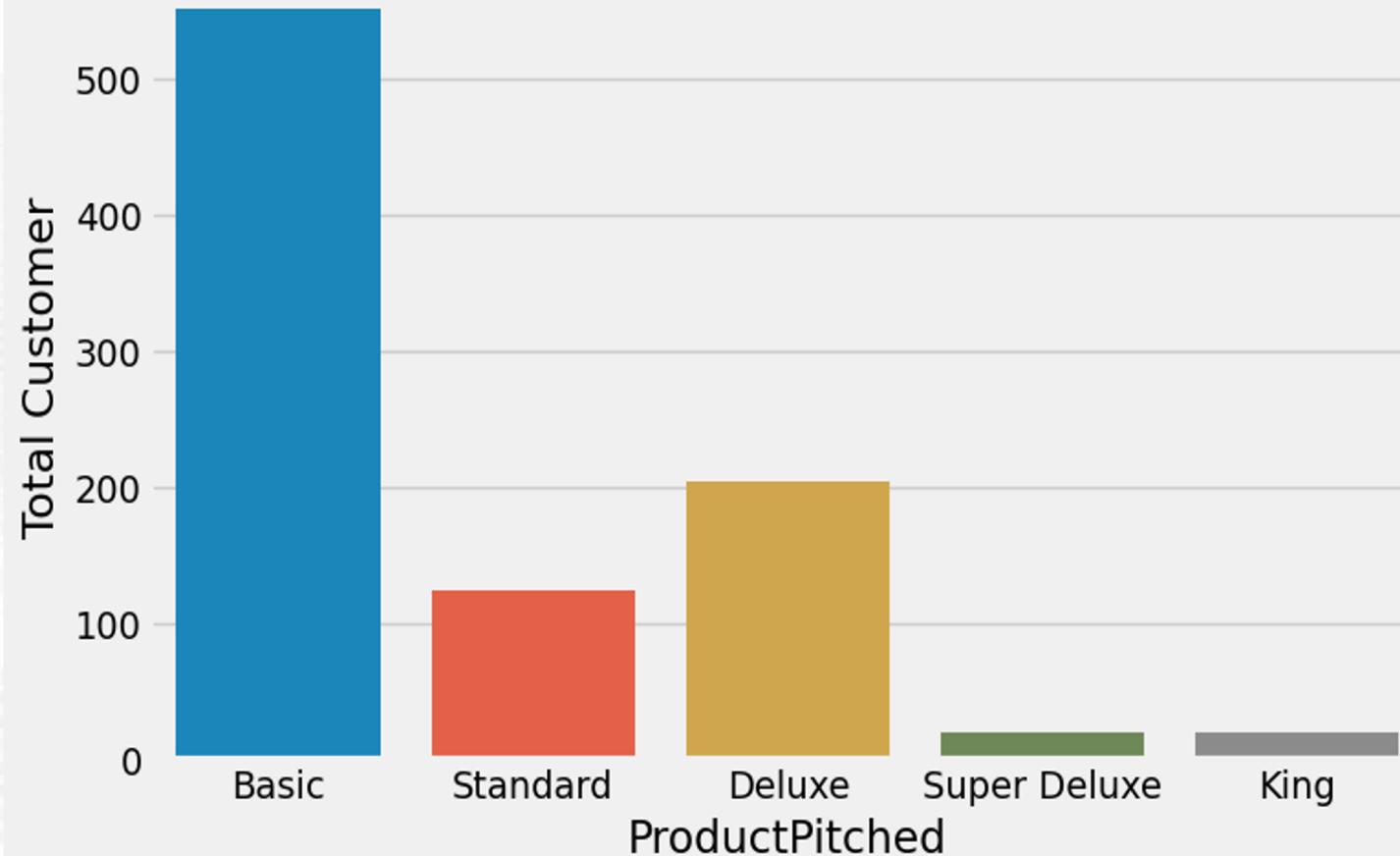
Action Items (Rekomendasi terhadap Insight)

- **Penargetan Pelanggan:**
Mendorong pelanggan untuk memperoleh paspor dengan melakukan penawaran khusus.

Pelanggan yang Menolak Tawaran Paket Liburan Didominasi oleh Pelanggan yang Tidak Memiliki Passport



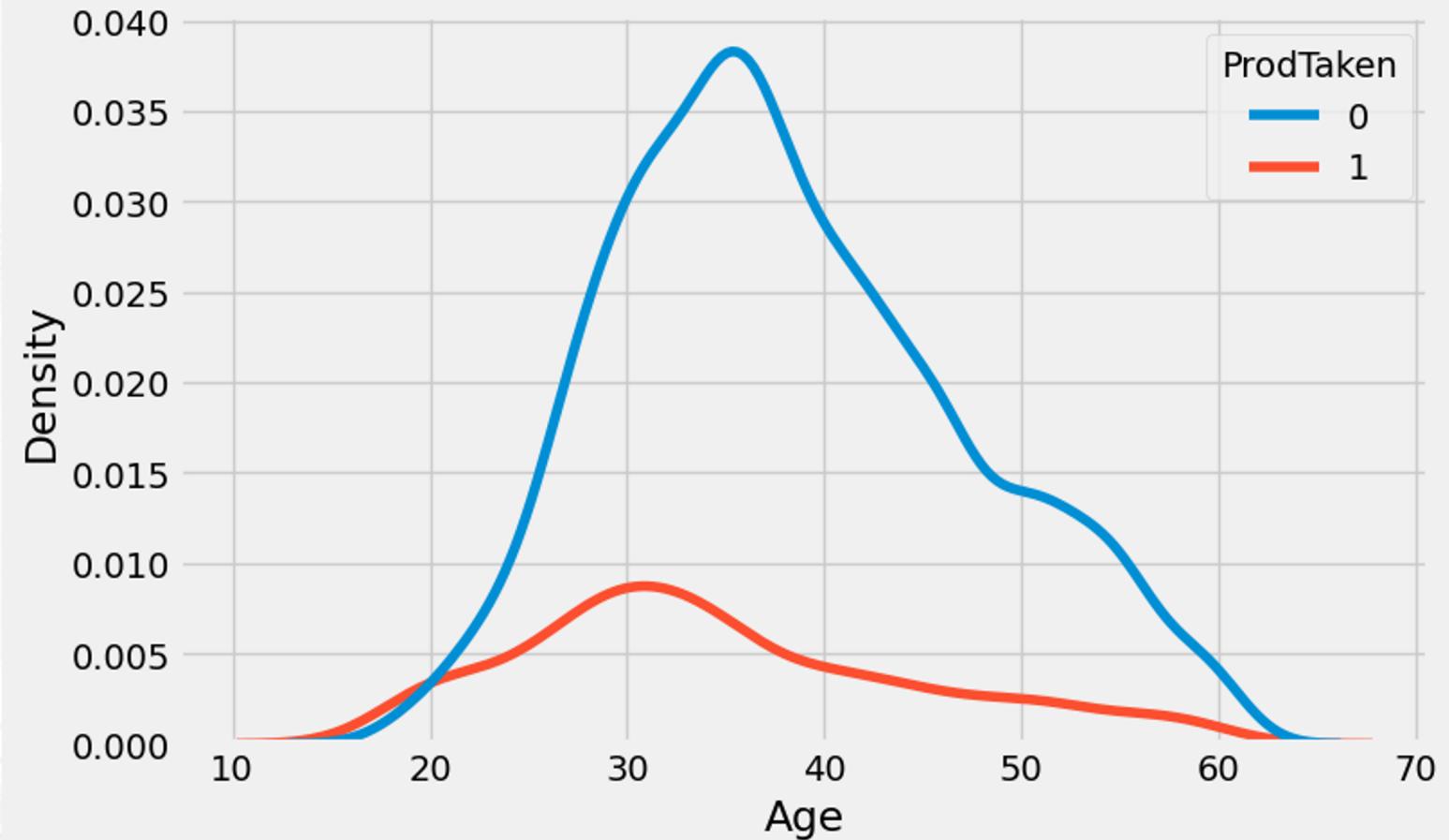
Pelanggan Cenderung Membeli Paket Kelas Rendah Hingga Paket Kelas Menengah (Basic - Deluxe)



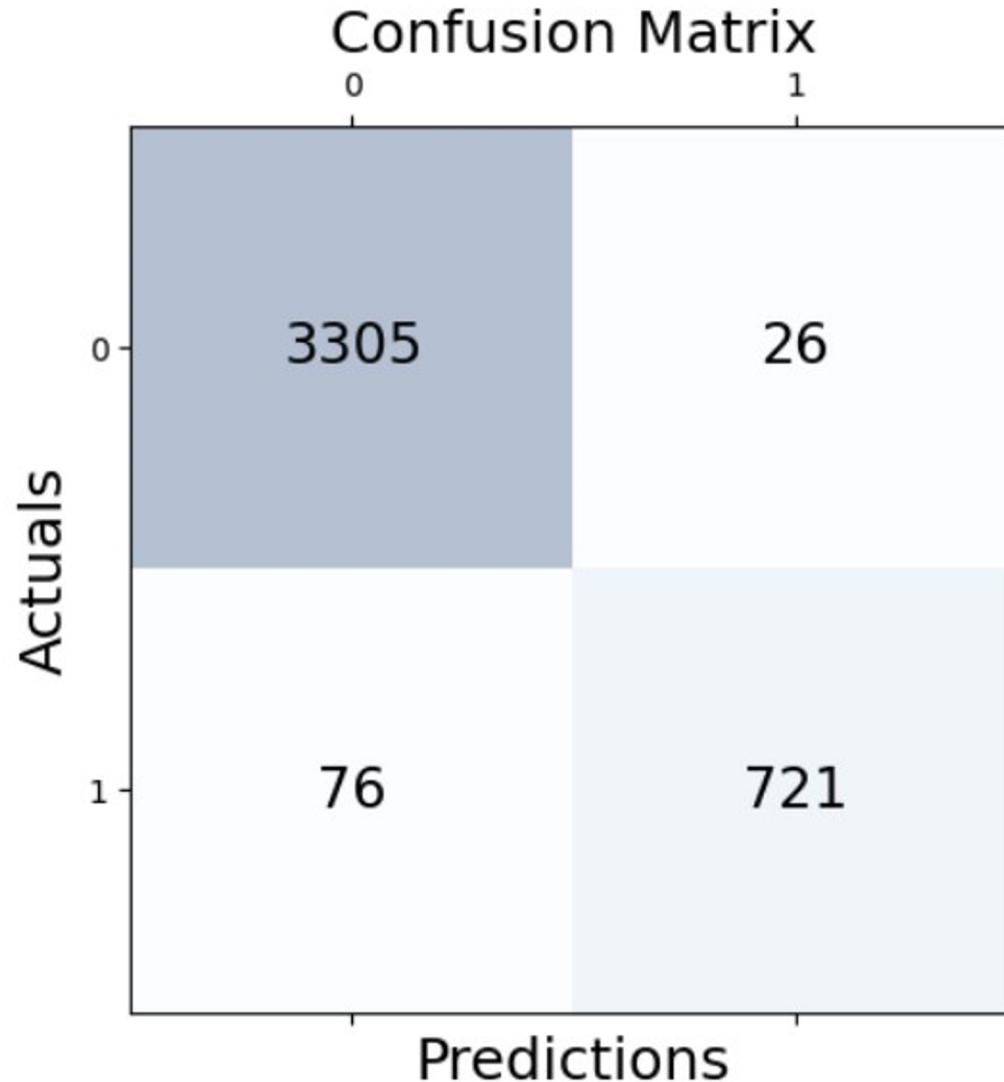
- **Penentuan Harga Paket Terbaru:** Perusahaan dapat menentukan harga paket terbaru menyesuaikan range harga paket kelas bawah hingga menengah (Basic, Standard, dan Deluxe), sehingga dapat menarik perhatian pelanggan.

- **Paket Liburan Terkini:**
Menawarkan paket liburan terkini yang mencakup tujuan dan destinasi wisata yang sedang diminati oleh kalangan usia muda ($\pm 20\text{-}35$ tahun).

Pelanggan yang Berumur Muda ($\pm 20\text{-}35$ tahun) Cenderung Lebih Membeli Tawaran Paket Liburan



Business Simulation



$TN = 3305$ orang

$FN = 76$ orang

$FP = 26$ orang

$TP = 721$ orang

Karena kami memilih metric precision, jadi kami memfokuskan pada **nilai FP dan TP**.

- FP adalah orang yang diberikan penawaran, namun tidak membeli paket.
- TP adalah orang yang diberikan penawaran dan membeli paket.

Business Simulation

Conversion Rate

$$TP = \frac{TP}{TP + FP} \times 100\%$$

Sebelum adanya machine learning,
conversion rate-nya hanya **19%**.

Setelah adanya machine learning,
conversion rate-nya menjadi **97%**.

Jadi, dengan machine learning,
perusahaan dapat menaikkan
conversion rate sebanyak **78%**.

```
# Cek banyaknya customer yg beli (1) dan menolak (0) paket sebelum adanya machine learning
df['ProdTaken'].value_counts()
```

```
0    3968
1    920
Name: ProdTaken, dtype: int64
```

```
#hitung conversion rate
customer_yg_beli = 920 #ProdTaken==1
customer_yg_nolak = 3968 #ProdTaken==0
```

```
conversion_rate = customer_yg_beli / (customer_yg_beli + customer_yg_nolak)
print('Conversion Rate (Before):', round(conversion_rate,2))
```

Conversion Rate (Before): 0.19

```
# Cek banyaknya customer yg beli (TP) dan menolak (FP) paket setelah adanya machine learning
TP = conf_matrix[1,1]
FP = conf_matrix[0,1]
```

```
print('TP:',TP,'FP:',FP)
```

TP: 721 FP: 26

```
#hitung conversion rate
customer_yg_beli = 721 #TP
customer_yg_nolak = 26 #FP
```

```
conversion_rate = customer_yg_beli / (customer_yg_beli + customer_yg_nolak)
print('Conversion Rate (After):', round(conversion_rate,2))
```

Conversion Rate (After): 0.97

Total Revenue

$$\text{Revenue} = (\text{TP} * \text{Profit per package}) - ((\text{TP} + \text{FP}) * \text{Cost per marketing})$$

Menghitung **total revenue** dengan asumsi biaya marketing per *pitch*-nya 1/10 dari keuntungan penjualan per-paketnya.
ex: keuntungan per-paket Rp1.000.000 maka, biaya marketing per-paketnya Rp100.000

Sebelum adanya machine learning,
total revenue-nya adalah
Rp 431.200.000,-.

Setelah adanya machine learning,
total revenue-nya menjadi
Rp 646.300.000,-.

Jadi, dengan machine learning,
perusahaan dapat menaikkan total
revenue sebanyak Rp 215.100.000,-.

```
# Total Revenue sebelum adanya machine learning
customer_yg_beli = 920 #ProdTaken==1
customer_yg_nolak = 3968 #ProdTaken==0
profit_per_package = 1000000
cost_per_marketing = 100000

revenue = (customer_yg_beli * profit_per_package) - ((customer_yg_beli+customer_yg_nolak)*cost_per_marketing)
print(revenue, 'Rupiah')
```

431200000 Rupiah

```
# Total Revenue sesudah adanya model
customer_yg_beli = 721 #TP
customer_yg_nolak = 26 #FP
profit_per_package = 1000000
cost_per_marketing = 100000

revenue = (customer_yg_beli * profit_per_package) - ((customer_yg_beli+customer_yg_nolak)*cost_per_marketing)
print(revenue, 'Rupiah')
```

646300000 Rupiah

PEMBAGIAN TUGAS

NAMA	STAGE 0	STAGE 1	STAGE 2	STAGE 3	STAGE 4
Edgar	Menyiapkan plan untuk diskusi selanjutnya	Business insight, multivariate analysis, merapikan google colab	Feature selection, class imbalance, merapikan google colab	Feature importance, modeling, moderator	Merevisi google colab
Faris	Fasilitator zoom	Git, fasilitator zoom dan recording, notulis diskusi	Menghapus data tidak relevan, mengubah tipe data, Git, recording	Bisnis insight, split data train & tes, notulis, recording	Fasilitator zoom dan recording
Jannisah	Notulis diskusi grup	Membuat PPT, business insight, update trello	Handling missing values, outliers, membuat PPT	Hyperparameter tuning, rekomendasi insight, membuat PPT	Membuat PPT presentasi
Jodhi	Membuat google colab	Univariate analysis, moderator, descriptive statistics	Handling missing values, outliers, notulis	Modeling, feature importance, notulis	Merevisi laporan final project

NAMA	STAGE 0	STAGE 1	STAGE 2	STAGE 3	STAGE 4
R. Arnanda	Monitoring	Notulis mentoring, descriptive statistics, moderator	Feature transformation, feature encoding, notulis	Bisnis insight, model evaluasi, notulis	Membuat business recommendation
Sendy	Menghubungi mentor, memimpin jalannya diskusi grup	Git, multivariate analysis, moderator	Menghapus data tidak relevan, mengubah tipe data, Git	Split data train & tes, rekomendasi insight, merapikan google colab	Notulis mentoring
Teguh	Moderator	Descriptive statistics, Git, notulis diskusi	Feature transformation, feature encoding, moderator	Hyperparameter tuning, merapikan google colab	Merevisi laporan final project
Vionella	Notulis mentoring	Merapikan google colab, membuat PPT, multivariate analysis	Feature selection, class imbalance, membuat PPT	Feature selection, model evaluasi, membuat PPT	Presentasi