Rakamin Academy

# Pentagon Trip
# Tour and Travel

Dokumen Laporan Final Project

# Stage 0

Preparation

# Stage 0 (Preparation)

## Role & Problem Statement

**Role:**

- Sebagai *Data Scientist*, kami bertanggung jawab menyelesaikan *problem* perusahaan berdasarkan data yang tersedia menggunakan berbagai macam teknik statistika, EDA, visualisasi data & *machine learning*

**Problem Statement:**

- PentagonTrip.com adalah perusahaan travel yang menjual paket travel
- Paket travel yang dijual ada 5 : 1) Basic 2) Standard 3) Deluxe 4) Super Deluxe 5) King
- PentagonTrip.com menghubungi, menawarkan & menjual paket travel melalui telemarketing
- Calon customer dihubungi secara random tanpa kriteria apapun
- Tahun lalu, dari total calon customer yang dihubungi, hanya 18% yang membeli paket
- Karena ini, efek dari biaya pemasaran kurang efisien & efektif sehingga *revenue* yang didapat kurang maksimal

**“Bagaimana caranya membantu tim marketing agar mendapatkan *revenue* yang maksimal & mengurangi *marketing cost* yang terbuang sia-sia?”**

# Stage 0 (Preparation)

## Goal, Objectives, & Business Metrics

**Goal:** Mengefisienkan *telemarketing cost* agar dapat memaksimalkan *revenue*

**Objectives:** Membuat model prediksi untuk menentukan target customer yang lebih potensial untuk membeli paket travel yang ada berdasarkan data yang tersedia

**Business Metrics:**

- *Revenue*: Peningkatan pendapatan yang diperoleh dari customer yang membeli paket travel
- *Spending Revenue on Telemarketing Cost*: Penurunan persentase / rasio biaya telemarketing terhadap pendapatan

***"Satu hal yang penting adalah tidak ada satu pun pemasar yang bisa menyasar semua segmen. Jika pun ada, pasti membutuhkan dana yang tidak sedikit dan upaya yang tidak biasa."*** - Sumber

Hermawan Kartajaya, Founder & Chairman MarkPlus, Inc.

# Stage 1

EDA, Insight, and Visualization

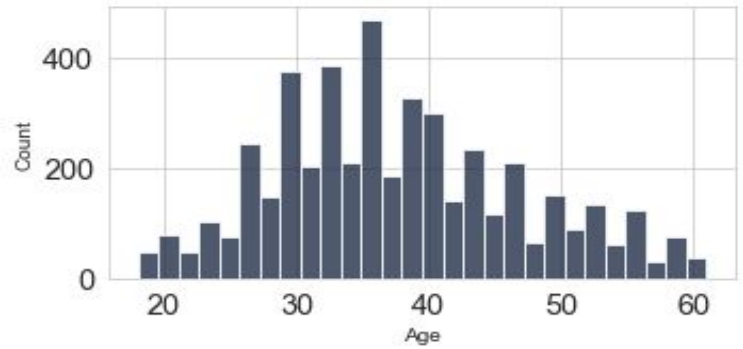
# Stage 1 (EDA, Insight, and Visualization)

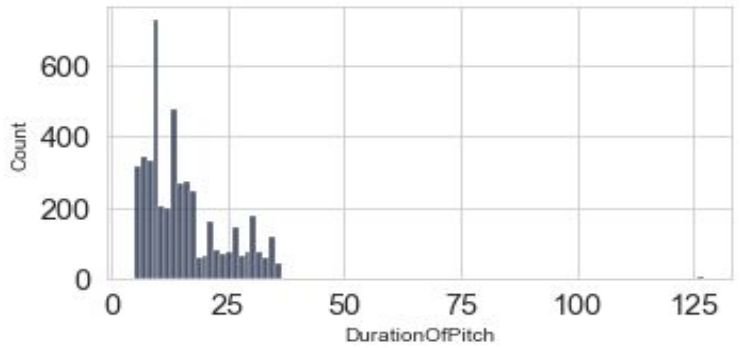
Feature

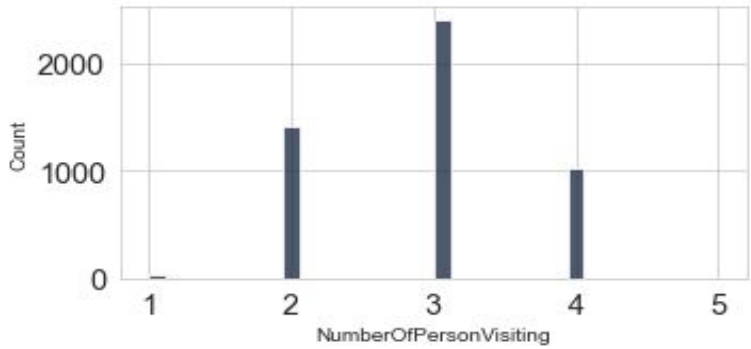
| No | Feature | Keterangan |
|---|---|---|
| 1. | *customerID* | Unique customer ID |
| 2. | *ProdTaken* (Target) | The customer taking the package or not (0: No, 1: Yes) |
| 3. | *Age: Age of customer* | Age of customer |
| 4. | *TypeofContract* | How customer was contacted (Company Invited or Self Inquiry) |
| 5. | *CityTier* | City tier depends on the development of a City,population,facilities,and living standards. the categories are ordered i.e |
| 6. | *DurationofPitch* | Duration of the pitch by a salesperson to the customer |
| 7. | *Occupation* | Occupation of customer |
| 8. | *Gender* | Gender of customer |
| 9. | *NumberOfPersonVisiting* | Total number of persons planning to take the trip with the customer |
| 10. | *NumberOfFollowups* | Total number of follow-ups has been done by the salesperson after the sales pitch |
| 11. | *ProductPitched* | Product pitched by the salesperson |
| 12. | *PreferredPropertyStar* | Preferred hotel property rating by customer |
| 13. | *MaritalStatus* | Marital status of customer |
| 14. | *NumberOfTrips* | Average number of trips in a year by customer |
| 15. | *Passport* | The customer has a passport or not (0: No, 1: Yes) |
| 16. | *PitchSatisfactionScore* | Sales pitch satisfaction score |
| 17. | *OwnCar* | Whether the customers own a car or not (0: No, 1: Yes) |
| 18. | *NumberOfChildrenVisiting* | Total number of children with age less than 5 planning to take the trip with the customer |
| 19. | *Designation* | Designation of the customer in the current organization |
| 20. | *MonthlyIncome* | Gross monthly income of the customer |

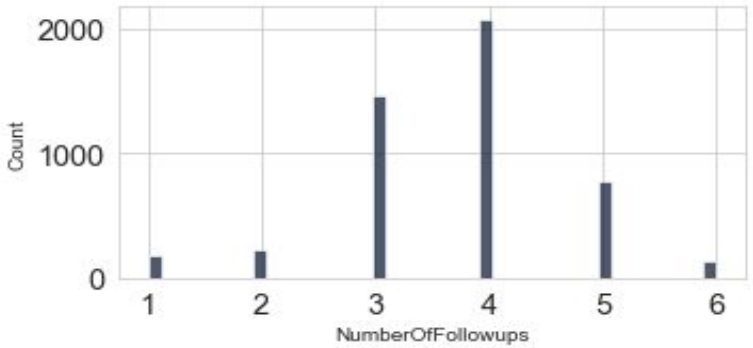
# Stage 1 (EDA, Insight, and Visualization)

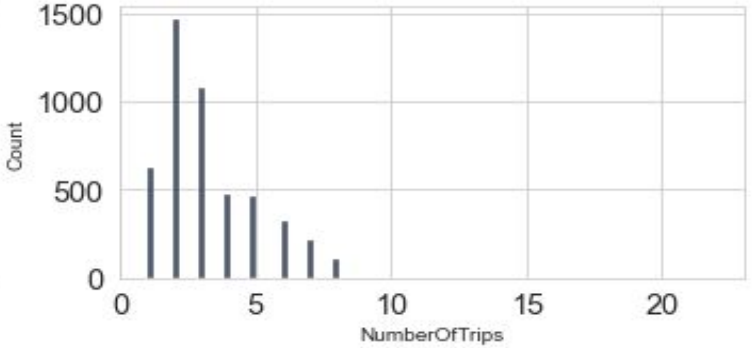
## Numerical Variables

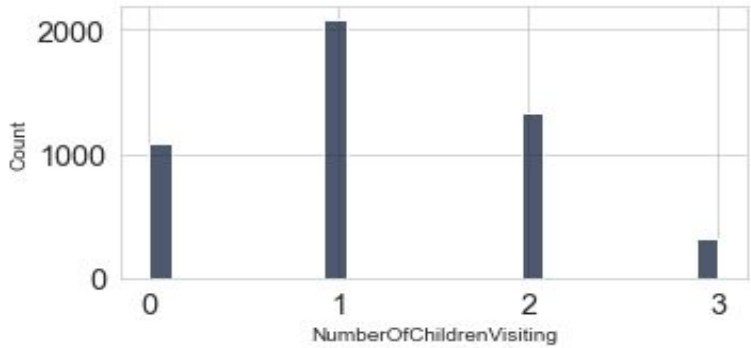


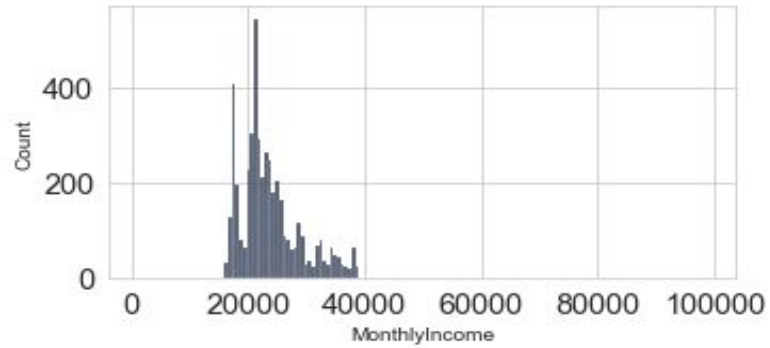



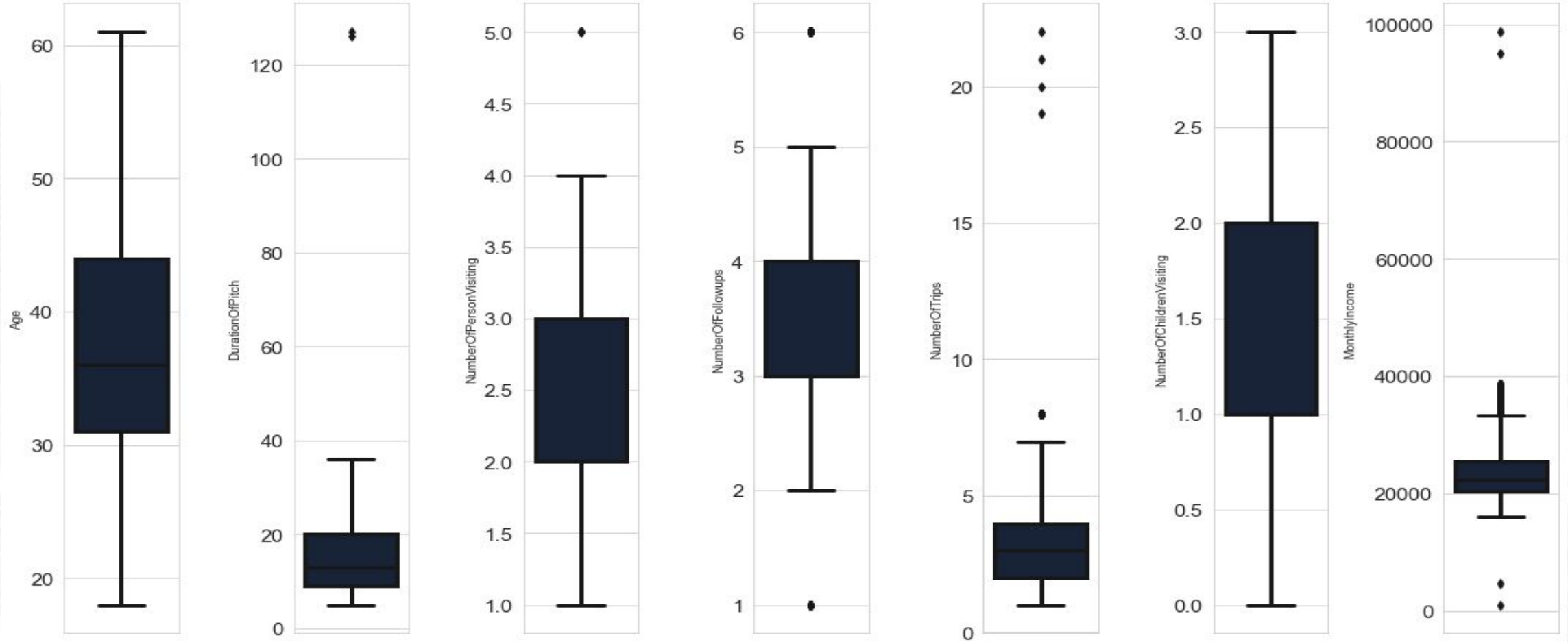




Distribusi pada feature Age, NumberOfPersonVisiting, NumberOfFollowups, dan NumberOfChildrenVisiting terlihat hampir normal

Distribusi pada feature DurationOfPitch, NumberOfTrips, dan MonthlyIncome terlihat right skewed

# Stage 1 (EDA, Insight, and Visualization)

## Numerical Variables

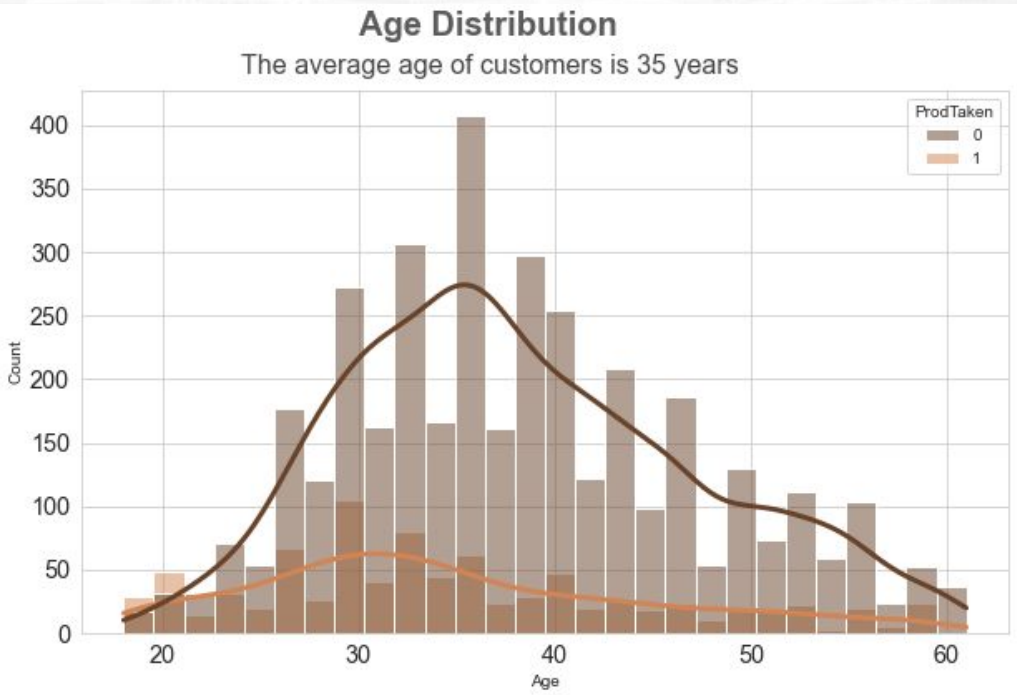



Feature yang tidak memiliki outlier yakni Age dan NumberOfChildrenVisiting, sedangkan yang lainnya memiliki outlier

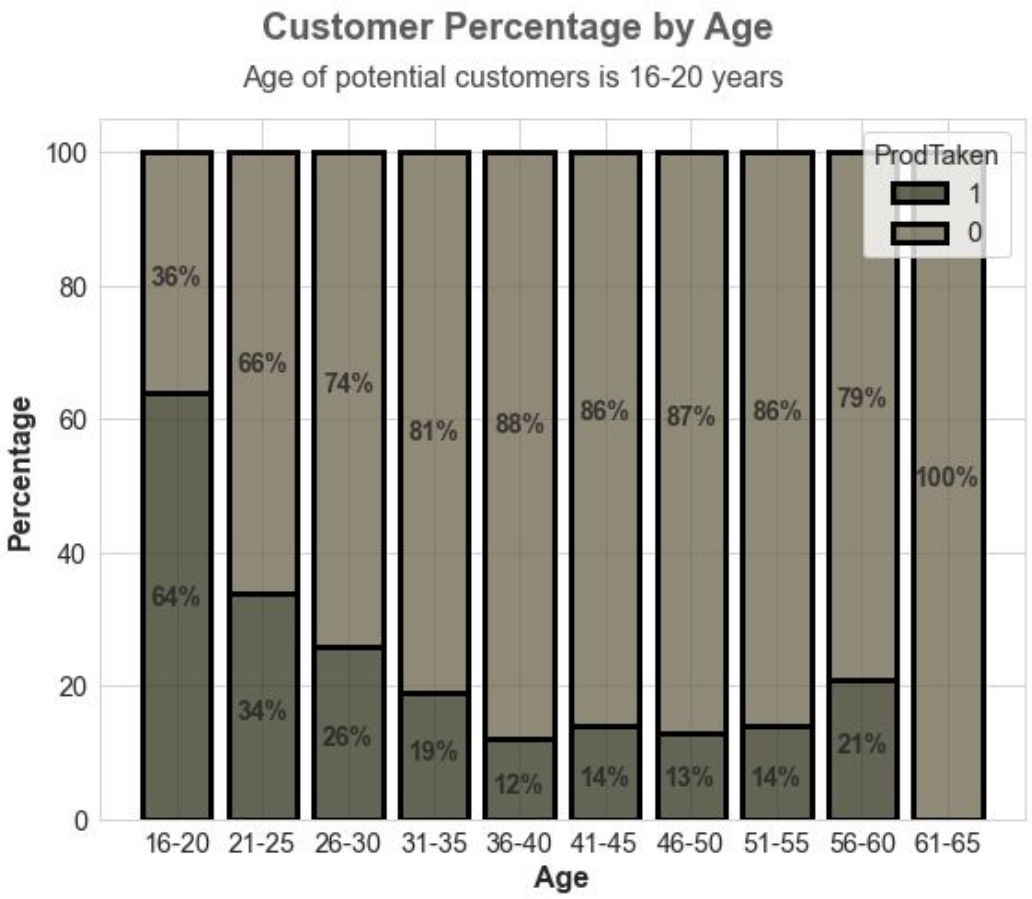
# Stage 1 (EDA, Insight, and Visualization)

## Numerical Variables

### Age




Rata-rata customer memiliki umur 35 tahun

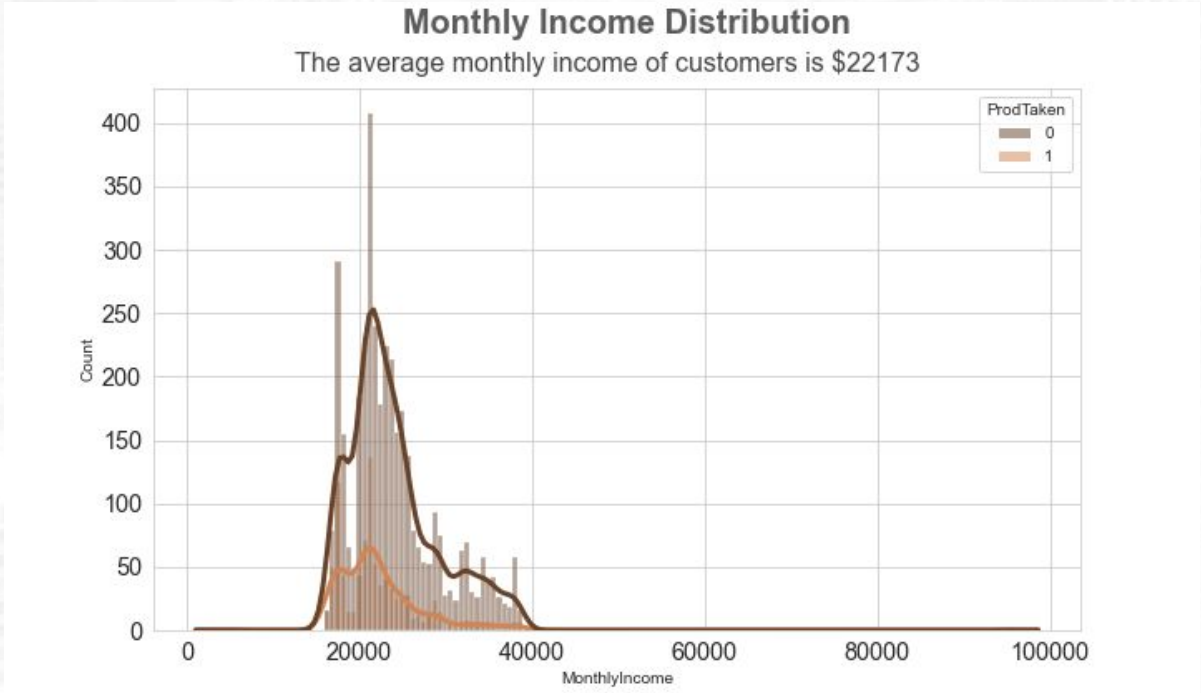



Customer dengan umur 16-20 tahun lebih berpotensi untuk mengambil paket

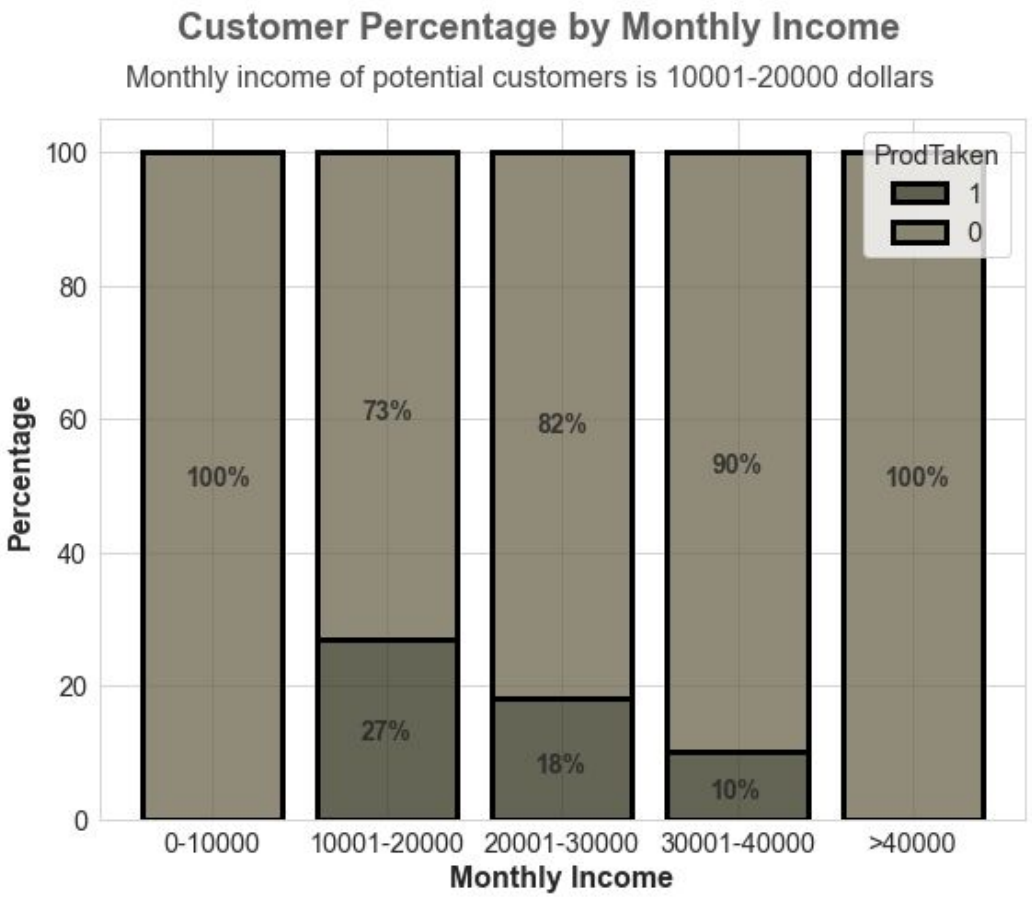
# Stage 1 (EDA, Insight, and Visualization)

## Numerical Variables

### Monthly Income




Rata-rata customer memiliki monthly income $22,173

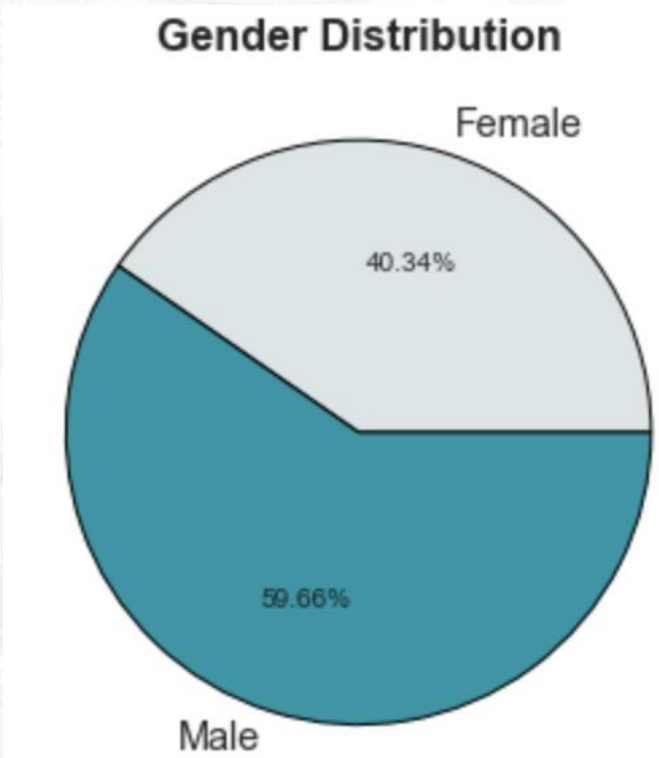



Customer dengan monthly income 10,001-20,000 dolar lebih berpotensi untuk mengambil paket

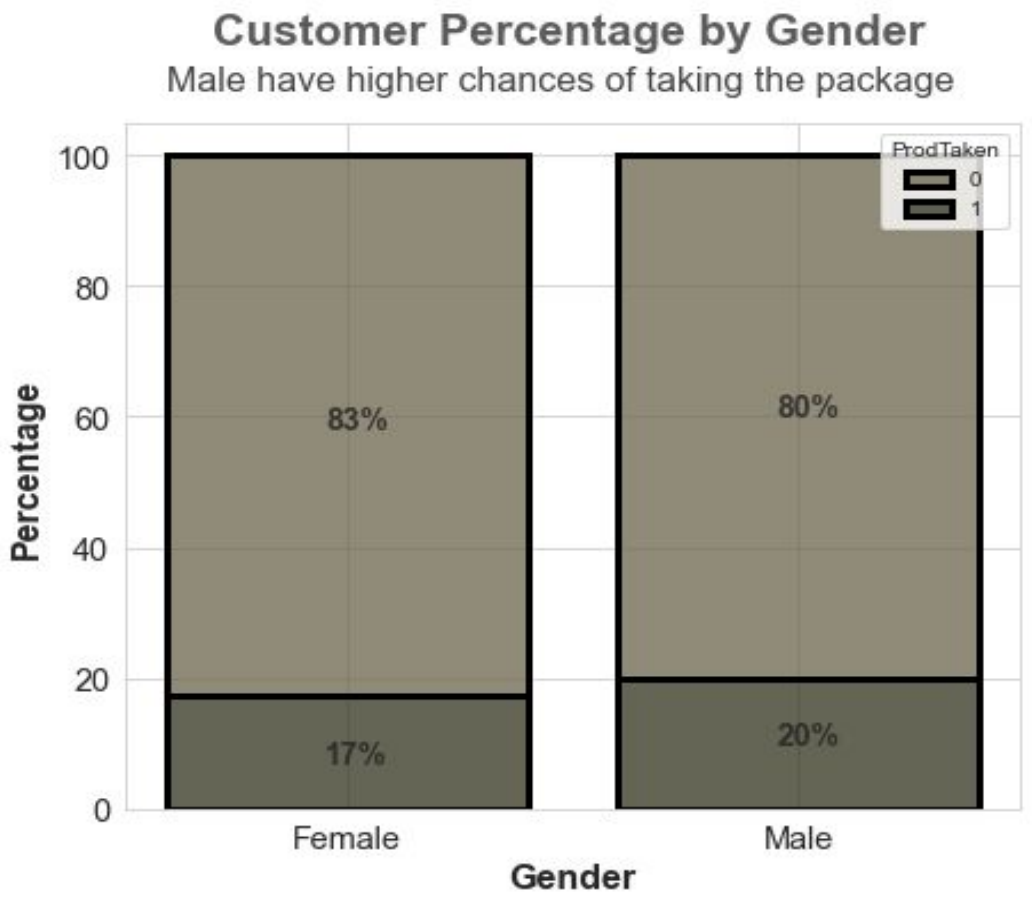
# Stage 1 (EDA, Insight, and Visualization)

## Categorical Variables

### Gender




Rata-rata customer bergender Male

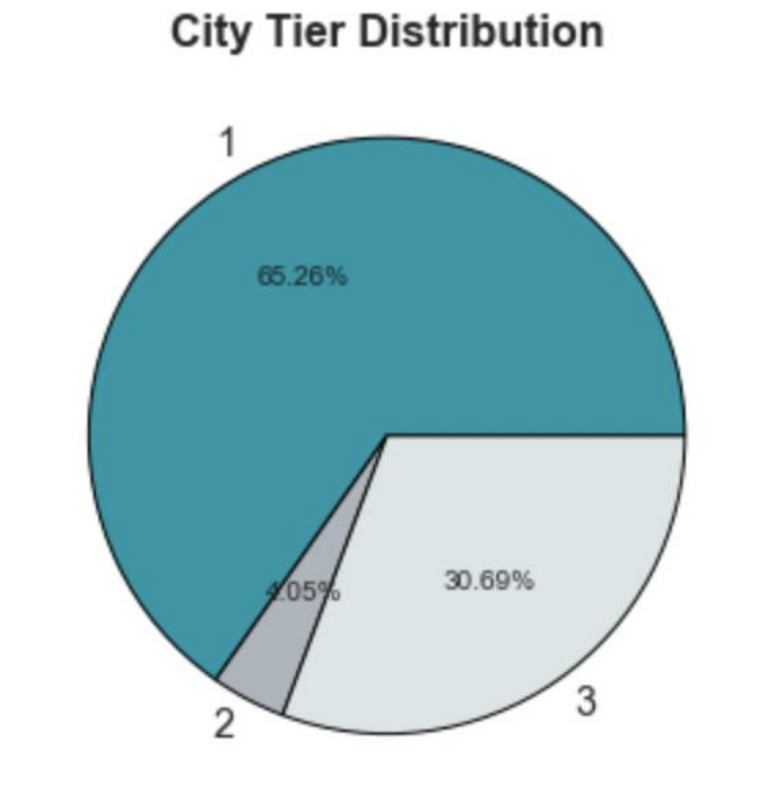



Customer dengan gender Male lebih berpotensi untuk mengambil paket

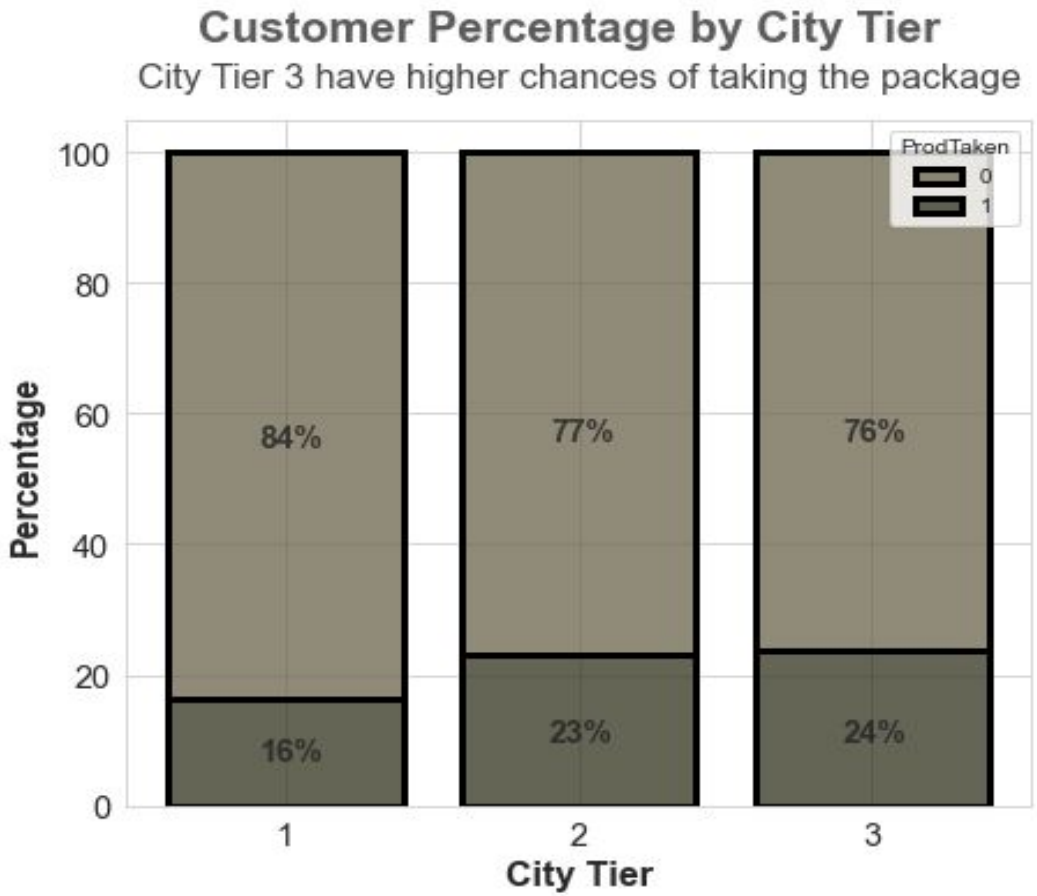
# Stage 1 (EDA, Insight, and Visualization)

## Categorical Variables

### City Tier




Rata-rata customer berada di city tier 1

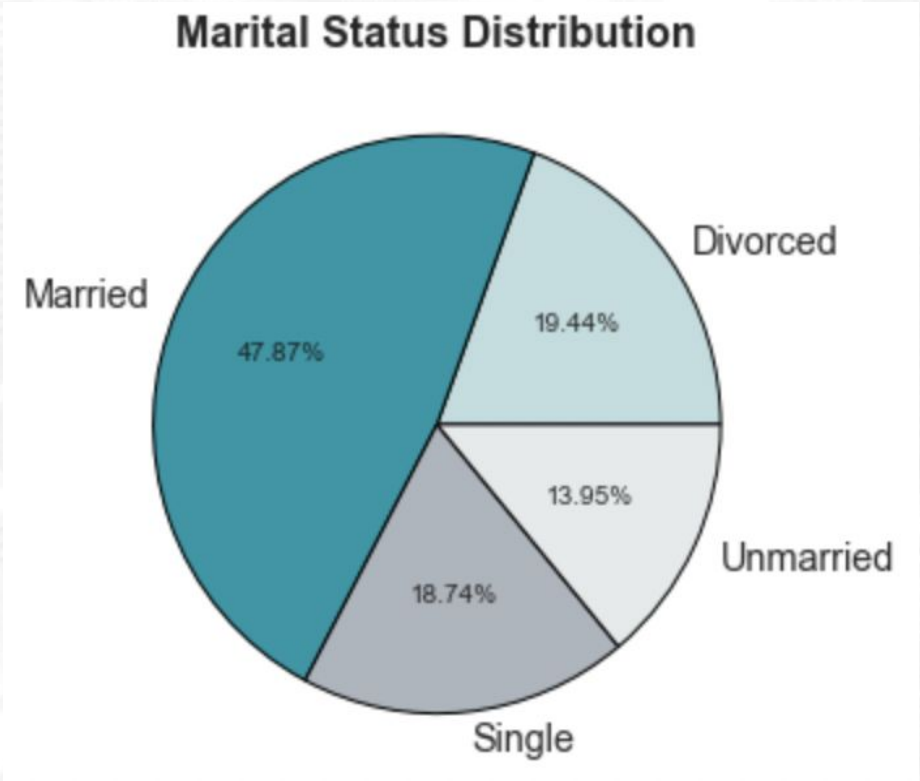



Customer dengan city tier 3 lebih berpotensi untuk mengambil paket

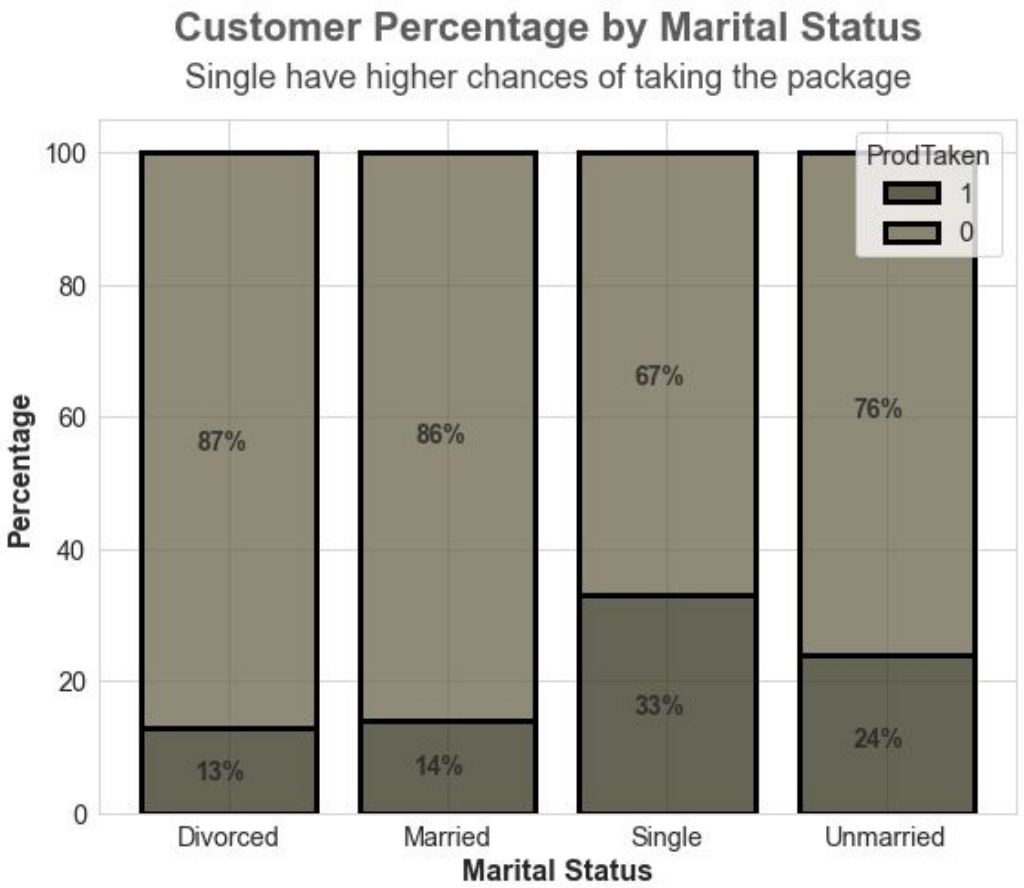
# Stage 1 (EDA, Insight, and Visualization)

## Categorical Variables

### Marital Status




Rata-rata customer berstatus married

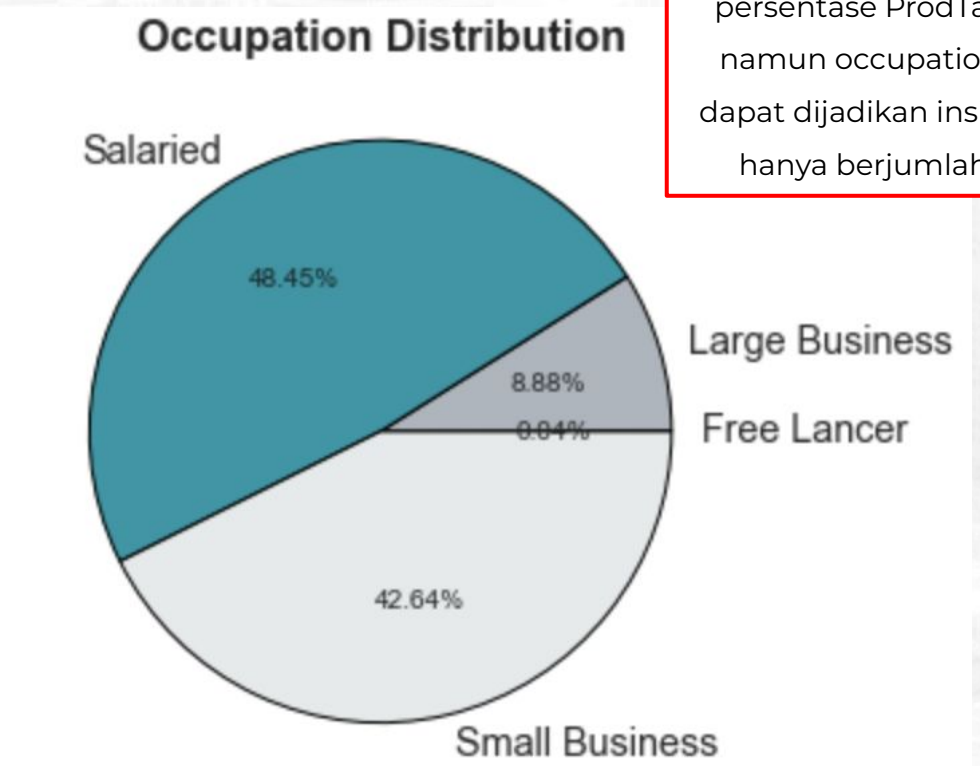



Customer dengan marital status Single lebih berpotensi untuk mengambil paket

# Stage 1 (EDA, Insight, and Visualization)

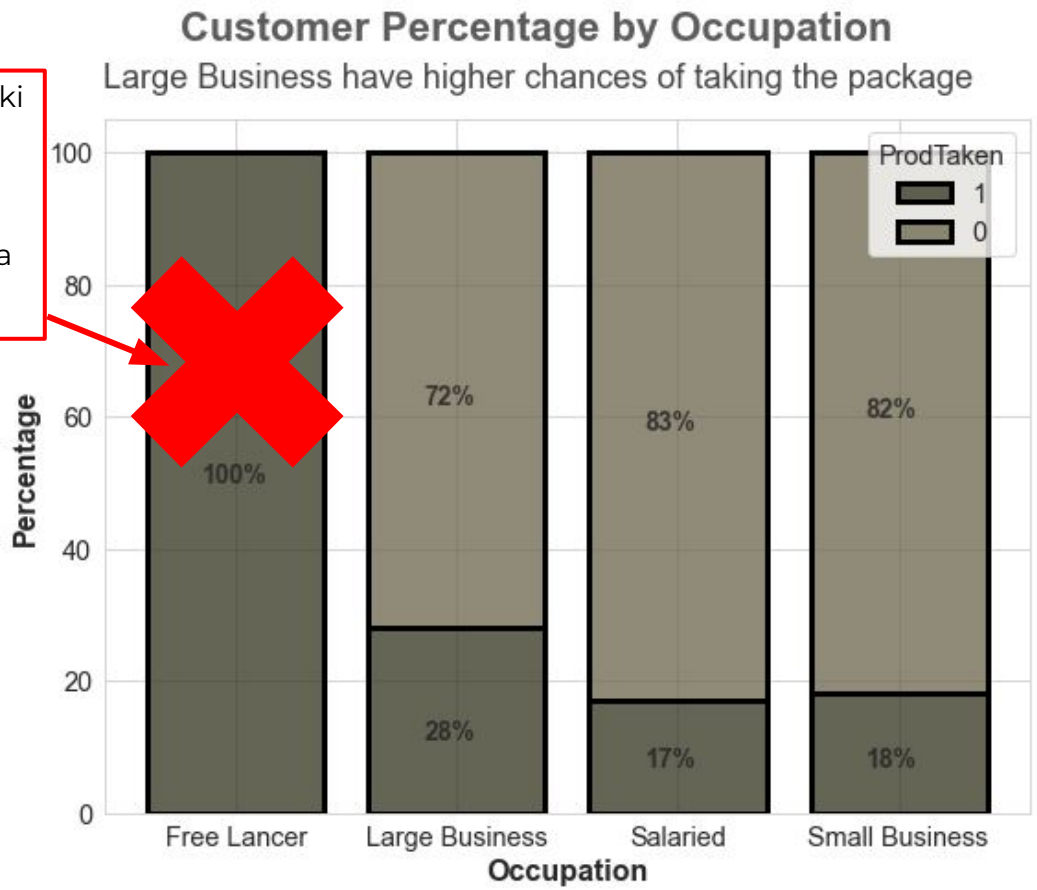
## Categorical Variables

### Occupation




Rata-rata customer memiliki occupation Salaried

Walaupun Free Lancer memiliki persentase ProdTaken 100%, namun occupation ini tidak dapat dijadikan insight karena hanya berjumlah 2 orang

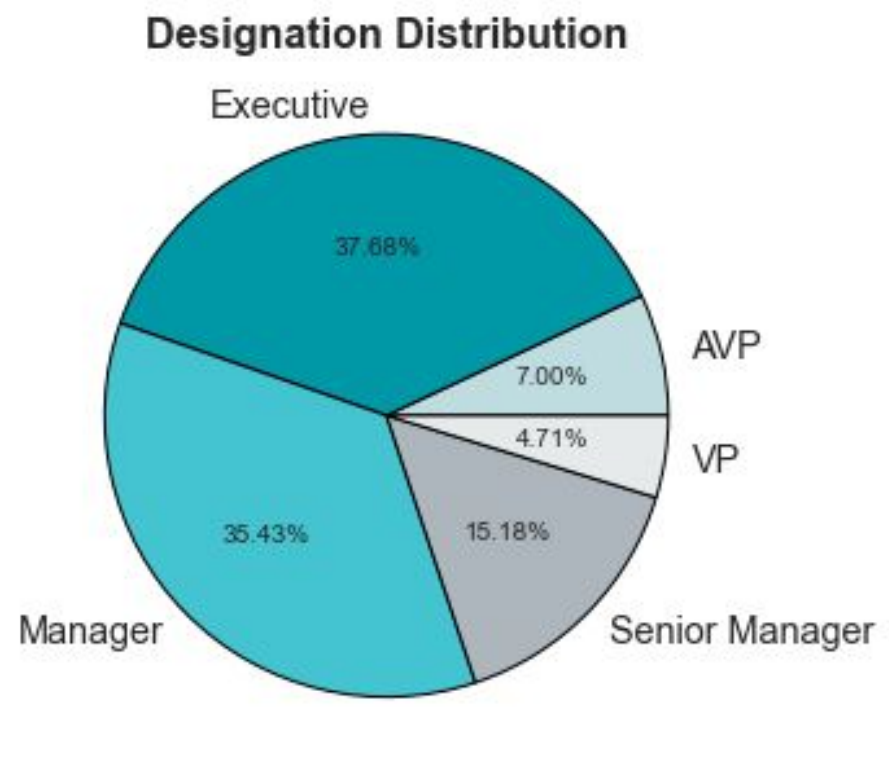



Customer dengan occupation Large Business lebih berpotensi untuk mengambil paket

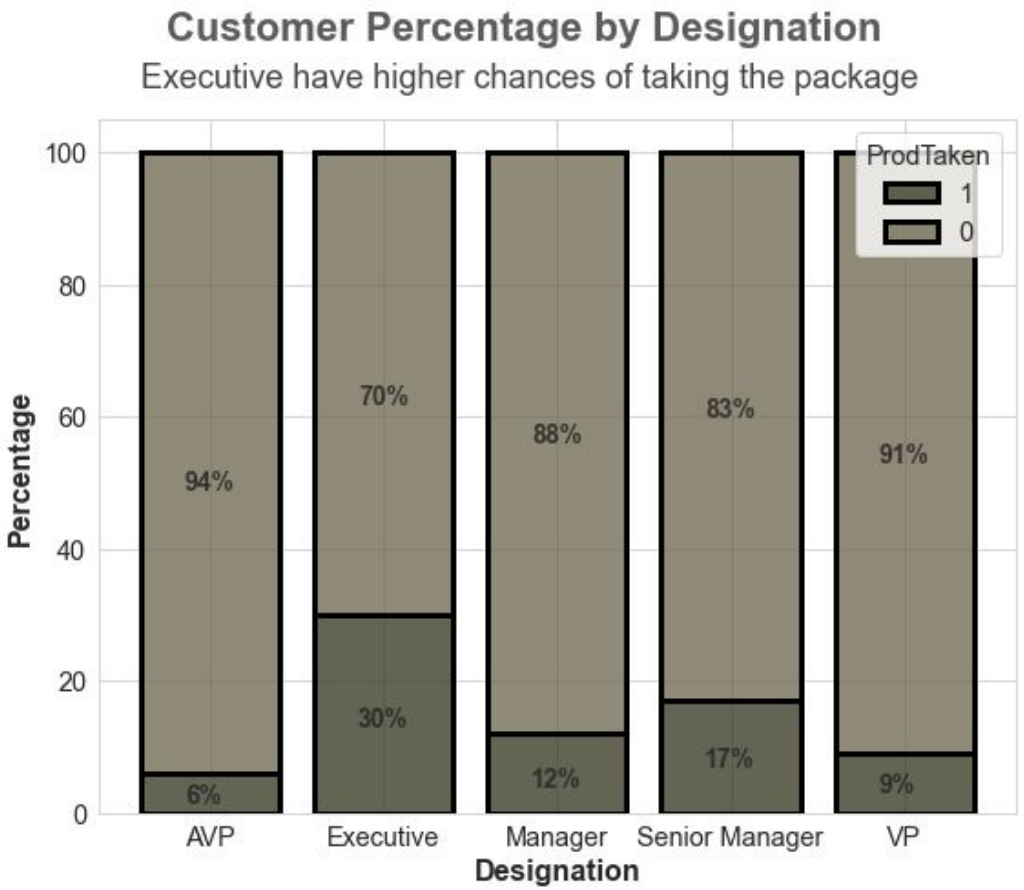
# Stage 1 (EDA, Insight, and Visualization)

## Categorical Variables

### Designation




Rata-rata customer memiliki designation Executive

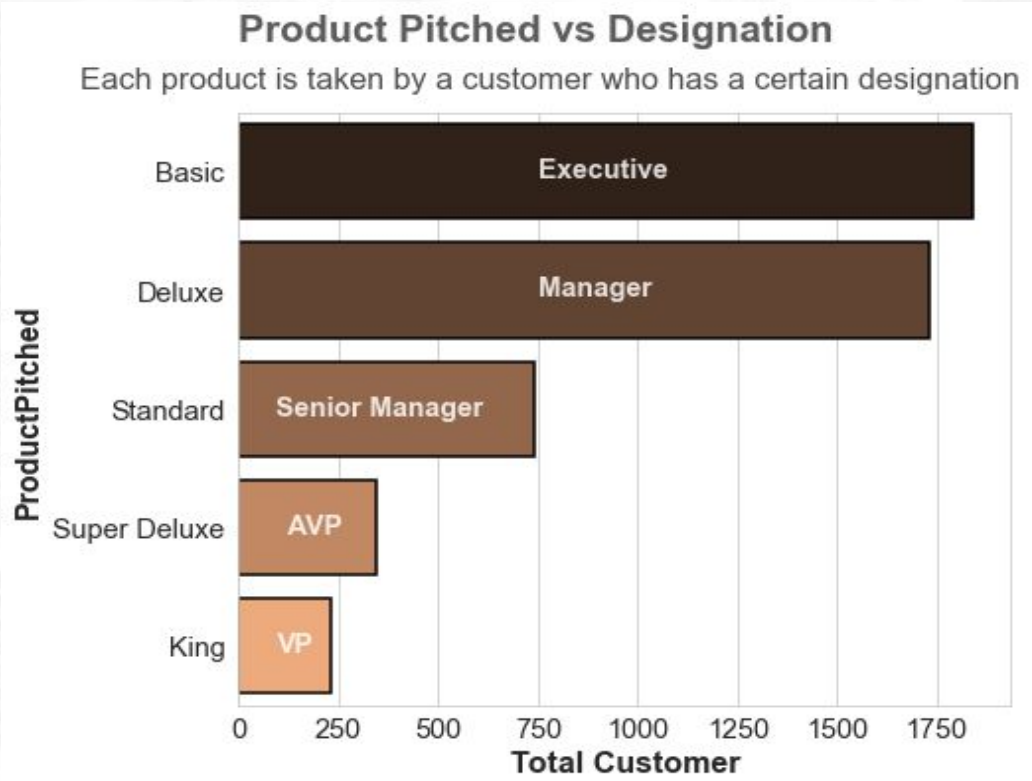



Customer dengan designation Executive lebih berpotensi untuk mengambil paket

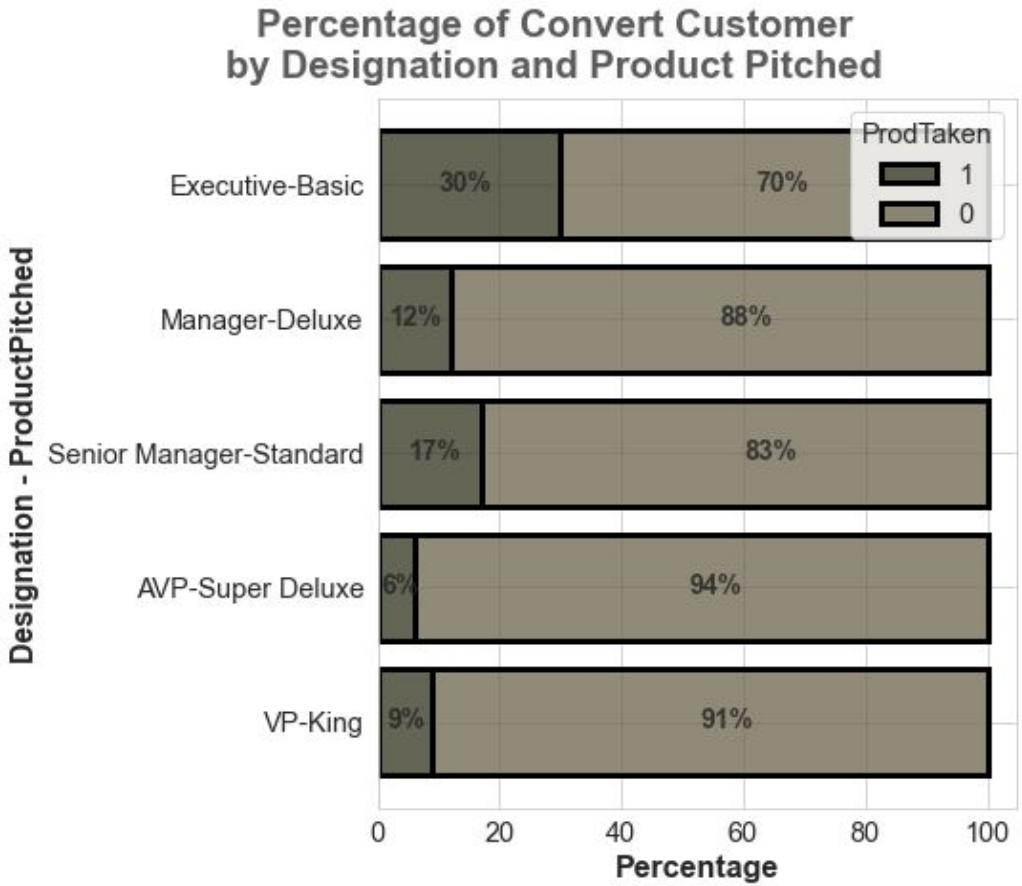
# Stage 1 (EDA, Insight, and Visualization)

## Categorical Variables

### Product Pitched vs Designation




Customer hanya ditawarkan paket tertentu berdasarkan Designationnya




Tidak banyak customer yang mengambil paket

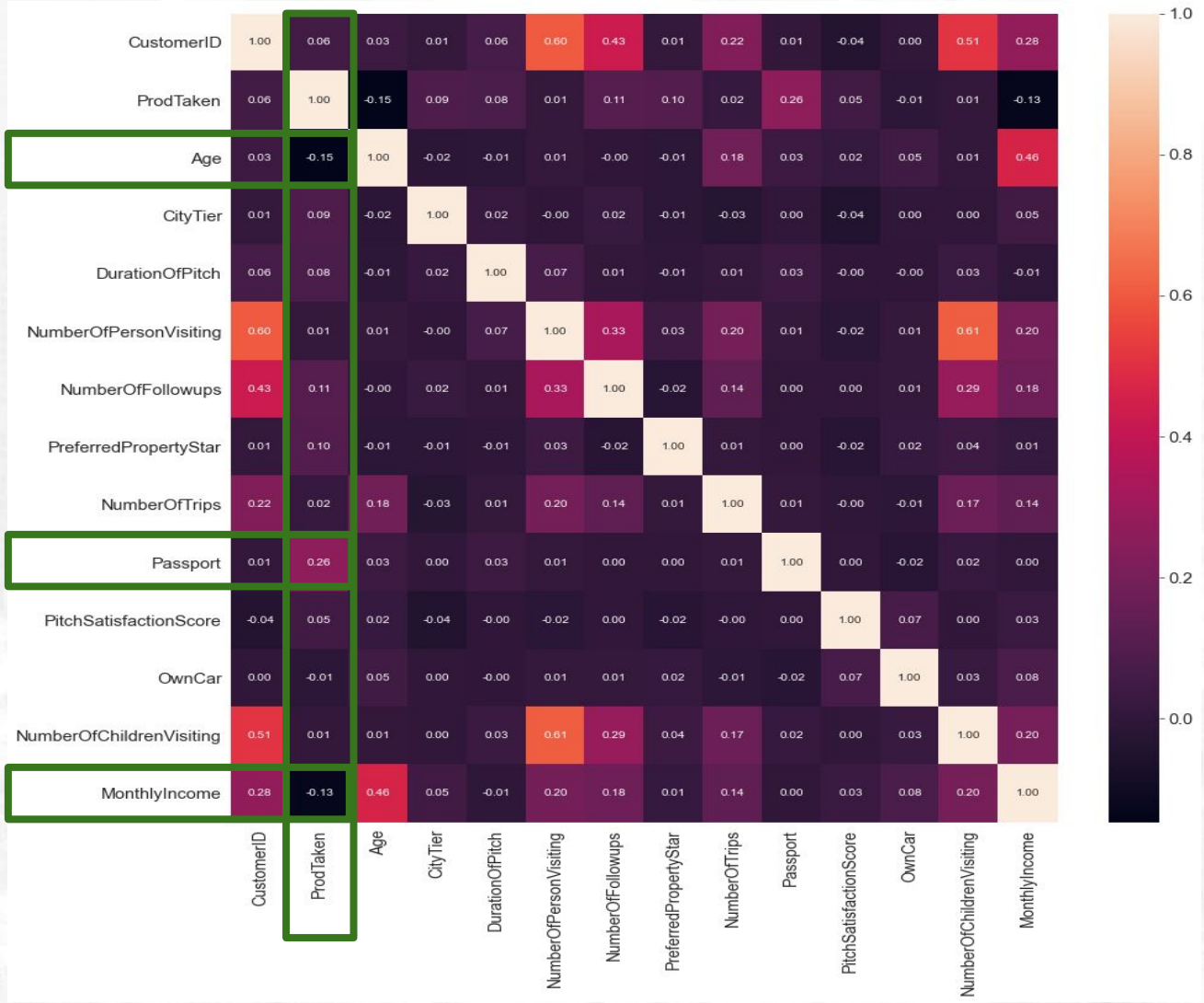
# Stage 1 (EDA, Insight, and Visualization)

## Heatmap

Dari correlation heatmap di samping dapat dilihat bahwa:

- Seluruh feature memiliki korelasi lemah dengan target
- Top 3 feature yang berkorelasi kuat dengan target: Passport, Age, dan MonthlyIncome
- Feature OwnCar, NumberOfChildrenVisiting, dan NumberOfPersonVisiting, memiliki korelasi yang paling lemah dengan target sehingga mungkin nantinya fitur-fitur ini dapat diabaikan
- Feature yang saling berkorelasi kuat/redundan (>0.3): NumberOfPersonVisiting dengan NumberOfChildrenVisiting, NumberOfPersonVisiting dengan NumberOfFollowups, dan Age dengan MonthlyIncome. Nantinya salah satu dari variabel yang berkorelasi akan didrop (akan dilihat dari nilai VIF (Variance Inflation Factor) nya).

Kemungkinan feature yang didrop: OwnCar, NumberOfChildrenVisiting, NumberOfPersonVisiting, dan MonthlyIncome.



# Stage 1 (EDA, Insight, and Visualization)

## EDA Conclusions

- Distribusi pada feature Age, NumberOfPersonVisiting, NumberOfFollowups, dan NumberOfChildrenVisiting terlihat hampir normal, sedangkan feature DurationOfPitch, NumberOfTrips, dan MonthlyIncome terlihat right skewed
- Feature yang tidak memiliki outlier yakni Age dan NumberOfChildrenVisiting, lainnya memiliki outlier
- Seluruh feature memiliki korelasi lemah dengan target
- Top 3 feature yang berkorelasi kuat dengan target: Passport, Age, dan MonthlyIncome
- Feature OwnCar, NumberOfChildrenVisiting, dan NumberOfPersonVisiting, memiliki korelasi yang paling lemah dengan target sehingga mungkin nantinya fitur-fitur ini dapat diabaikan
- Feature yang saling berkorelasi kuat/redundan (>0.3): NumberOfPersonVisiting dengan NumberOfChildrenVisiting, NumberOfPersonVisiting dengan NumberOfFollowups, dan Age dengan MonthlyIncome. Nantinya salah satu dari variabel yang berkorelasi akan didrop (akan dilihat dari nilai VIF (Variance Inflation Factor) nya).
- Kemungkinan feature yang didrop: OwnCar, NumberOfChildrenVisiting, NumberOfPersonVisiting, dan MonthlyIncome

# Stage 1 (EDA, Insight, and Visualization)

## Business Insight Conclusion

- Customer dengan monthly income 10001-20000 dolar lebih berpotensi untuk mengambil paket
- Customer dengan umur 16-20 tahun lebih berpotensi untuk mengambil paket
- Customer terbanyak :
    - Gender : Male
    - City : Tier 1
    - Marital Status : Married
    - Occupation : Salaried
    - Designation : Executive
- Customer potensial :
    - Gender : Male
    - City : Tier 3
    - Marital Status : Single
    - Occupation : Large Business
    - Designation : Executive

# Stage 2

## Data Preprocessing

# Stage 2 (Data Preprocessing)

## Merubah Tipe Data

Terdapat beberapa fitur dengan tipe data yang kurang tepat sehingga harus diubah terlebih dahulu.

**Code:**

**int64 → object**

```
df['ProdTaken'] = df['ProdTaken'].astype('str')
df['CityTier'] = df['CityTier'].astype('str')
df['Passport'] = df['Passport'].astype('str')
df['PitchSatisfactionScore'] = df['PitchSatisfactionScore'].astype('str')
df['OwnCar'] = df['OwnCar'].astype('str')
```

**float4 → object**

```
df['PreferredPropertyStar'] = df['PreferredPropertyStar'].astype('str')
```

**Before**

```
<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
```

**After**

```
<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   object
 2   Age                       4662 non-null   float64
 3   TypeofContact             4863 non-null   object
 4   CityTier                  4888 non-null   object
 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     4888 non-null   object
 12  MaritalStatus             4888 non-null   object
 13  NumberOfTrips             4748 non-null   float64
 14  Passport                  4888 non-null   object
 15  PitchSatisfactionScore    4888 non-null   object
 16  OwnCar                    4888 non-null   object
 17  NumberOfChildrenVisiting  4822 non-null   float64
 18  Designation               4888 non-null   object
 19  MonthlyIncome             4655 non-null   float64
dtypes: float64(6), int64(2), object(12)
memory usage: 763.9+ KB
```

# Stage 2 (Data Preprocessing)

## Mengelompokkan Data Kategorik dan Numerik

Setelah mengubah tipe data, dilakukan pengelompokan variabel kategorik dan numerik untuk mempermudah komputasi selanjutnya.

**Code:**

**Fitur Numerik**

nums =
['CustomerID','Age','DurationOfPitch','NumberOfPersonVisiting','NumberOfFollowups','NumberOfTrips','NumberOfChildrenVisiting','MonthlyIncome']

**Fitur Kategorik**

cats =
['TypeofContact','Occupation','Gender','ProductPitched','MaritalStatus','Designation', 'ProdTaken', 'CityTier', 'PreferredPropertyStar', 'Passport','PitchSatisfactionScore','OwnCar']

df[nums].describe()

| | CustomerID | Age | DurationOfPitch | NumberOfPersonVisiting | NumberOfFollowups | NumberOfTrips | NumberOfChildrenVisiting | MonthlyIncome |
|---|---|---|---|---|---|---|---|---|
| count | 4888.000000 | 4662.000000 | 4637.000000 | 4888.000000 | 4843.000000 | 4748.000000 | 4822.000000 | 4655.000000 |
| mean | 202443.500000 | 37.622265 | 15.490835 | 2.905074 | 3.708445 | 3.236521 | 1.187267 | 23619.853491 |
| std | 1411.188388 | 9.316387 | 8.519643 | 0.724891 | 1.002509 | 1.849019 | 0.857861 | 5380.698361 |
| min | 200000.000000 | 18.000000 | 5.000000 | 1.000000 | 1.000000 | 1.000000 | 0.000000 | 1000.000000 |
| 25% | 201221.750000 | 31.000000 | 9.000000 | 2.000000 | 3.000000 | 2.000000 | 1.000000 | 20346.000000 |
| 50% | 202443.500000 | 36.000000 | 13.000000 | 3.000000 | 4.000000 | 3.000000 | 1.000000 | 22347.000000 |
| 75% | 203665.250000 | 44.000000 | 20.000000 | 3.000000 | 4.000000 | 4.000000 | 2.000000 | 25571.000000 |
| max | 204887.000000 | 61.000000 | 127.000000 | 5.000000 | 6.000000 | 22.000000 | 3.000000 | 98678.000000 |

df[cats].describe()

| | TypeofContact | Occupation | Gender | ProductPitched | MaritalStatus | Designation | ProdTaken | CityTier | PreferredPropertyStar | Passport | PitchSatisfactionScore | OwnCar |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 4863 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 |
| unique | 2 | 4 | 3 | 5 | 4 | 5 | 2 | 3 | 4 | 2 | 5 | 2 |
| top | Self Enquiry | Salaried | Male | Basic | Married | Executive | 0 | 1 | 3.0 | 0 | 3 | 1 |
| freq | 3444 | 2368 | 2916 | 1842 | 2340 | 1842 | 3968 | 3190 | 2993 | 3466 | 1478 | 3032 |

# Stage 2 (Data Preprocessing)

## Memperbaiki Fitur Gender

Terdapat kesalahan value pada fitur gender, yakni ‘Fe Male’ sehingga terdapat 3 unique values pada fitur.
Dilakukan pengubahan values menjadi ‘Female’ sehingga hanya terdapat 2 unique values pada fitur.

**Code:**

```
df.replace(to_replace=r'Fe Male', value='Female', regex=True, inplace=True )
```

```
df[cats].describe()
```

**Before**

| | TypeofContact | Occupation | Gender | ProductPitched | MaritalStatus | Designation | ProdTaken | CityTier | PreferredPropertyStar | Passport | PitchSatisfactionScore | OwnCar |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 4863 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 |
| unique | 2 | 4 | 3 | 5 | 4 | 5 | 2 | 3 | 4 | 2 | 5 | 2 |
| top | Self Enquiry | Salaried | Male | Basic | Married | Executive | 0 | 1 | 3.0 | 0 | 3 | 1 |
| freq | 3444 | 2368 | 2916 | 1842 | 2340 | 1842 | 3968 | 3190 | 2993 | 3466 | 1478 | 3032 |

**After**

| | TypeofContact | Occupation | Gender | ProductPitched | MaritalStatus | Designation | ProdTaken | CityTier | PreferredPropertyStar | Passport | PitchSatisfactionScore | OwnCar |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 4863 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 | 4888 |
| unique | 2 | 4 | 2 | 5 | 4 | 5 | 2 | 3 | 4 | 2 | 5 | 2 |
| top | Self Enquiry | Salaried | Male | Basic | Married | Executive | 0 | 1 | 3.0 | 0 | 3 | 1 |
| freq | 3444 | 2368 | 2916 | 1842 | 2340 | 1842 | 3968 | 3190 | 2993 | 3466 | 1478 | 3032 |

# Stage 2 (Data Preprocessing)

## Handling Missing Values

Terdapat beberapa missing values. Terdapat fitur yang kami drop dan ada juga yang kami isi dengan median/modus.

**Code:**

**Drop → sulit untuk menentukan value yang cocok untuk mengisi data yang kosong. Selain itu juga karena data missing hanya sedikit**

```
df.dropna(inplace=True, subset=['TypeofContact','NumberOfChildrenVisiting'])
```

**Filling with Median → distribusi fitur skew sehingga diisi dengan median yang lebih robust terhadap outlier**

```
df['DurationOfPitch'].fillna(df['DurationOfPitch'].median(), inplace=True)
df['NumberOfFollowups'].fillna(df['NumberOfFollowups'].median(), inplace=True)
df['MonthlyIncome'].fillna(df['MonthlyIncome'].median(), inplace=True)
```

**Filling with Mode → diisi dengan mayoritas umur/jumlah trips customers**

```
df['Age'].fillna(df['Age'].mode()[0], inplace=True)
df['NumberOfTrips'].fillna(df['NumberOfTrips'].mode()[0], inplace=True)
```

df.isna().sum()

**Before**

```
CustomerID                  0
ProdTaken                   0
Age                       226
TypeofContact              25
CityTier                    0
DurationOfPitch           251
Occupation                  0
Gender                      0
NumberOfPersonVisiting      0
NumberOfFollowups          45
ProductPitched              0
PreferredPropertyStar       0
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
```

# Stage 2 (Data Preprocessing)

## Duplicated Data

**Code:**

```
df.duplicated().sum()
```

**Output:**

0

Tidak ada data duplikat.

## Handling Outliers

**Code:**

```
from scipy import stats
print(f'Jumlah baris sebelum memfilter outlier: {len(df)}')
filtered_entries = np.array([True] * len(df))
for col in
['DurationOfPitch','NumberOfPersonVisiting','NumberOfFollowups','NumberOfTrips','MonthlyIncome']:
    zscore = abs(stats.zscore(df[col])) # hitung absolute z-scorenya
    filtered_entries = (zscore < 3) & filtered_entries

df = df[filtered_entries]
print(f'Jumlah baris setelah memfilter outlier: {len(df)}')
```

Jumlah baris sebelum memfilter outlier: **4797**

Jumlah baris setelah memfilter outlier: **4787**

Digunakan Z-score karena terlalu banyak data yang harus didrop jika menggunakan kriteria outlier IQR.

# Stage 2 (Data Preprocessing)

## Standardization

**Code:**

```
from sklearn.preprocessing import StandardScaler
df['NumberOfTrips_std'] = StandardScaler().fit_transform(df['NumberOfTrips'].values.reshape(len(df), 1))
df['MonthlyIncome_std'] = StandardScaler().fit_transform(df['MonthlyIncome'].values.reshape(len(df), 1))
df['DurationOfPitch_std'] = StandardScaler().fit_transform(df['DurationOfPitch'].values.reshape(len(df), 1))
df.drop(columns=['DurationOfPitch','NumberOfTrips','MonthlyIncome'], inplace=True)
```

Dilakukan standarisasi data pada fitur-fitur yang belum berdistribusi normal sehingga memiliki mean = 0 dan standard deviation = 1

**Before**

| | CustomerID | Age | DurationOfPitch | NumberOfPersonVisiting | NumberOfFollowups | NumberOfTrips | NumberOfChildrenVisiting | MonthlyIncome |
|---|---|---|---|---|---|---|---|---|
| count | 4787.000000 | 4787.000000 | 4787.000000 | 4787.000000 | 4787.000000 | 4787.000000 | 4787.000000 | 4787.000000 |
| mean | 202445.630457 | 37.396490 | 15.337790 | 2.906204 | 3.710257 | 3.176729 | 1.190934 | 23382.838730 |
| std | 1412.764584 | 9.066171 | 8.017699 | 0.725755 | 0.998026 | 1.765771 | 0.858025 | 4892.611455 |
| min | 200000.000000 | 18.000000 | 5.000000 | 1.000000 | 1.000000 | 1.000000 | 0.000000 | 16009.000000 |
| 25% | 201225.500000 | 31.000000 | 9.000000 | 2.000000 | 3.000000 | 2.000000 | 1.000000 | 20447.500000 |
| 50% | 202446.000000 | 36.000000 | 13.000000 | 3.000000 | 4.000000 | 3.000000 | 1.000000 | 22222.000000 |
| 75% | 203673.500000 | 43.000000 | 19.000000 | 3.000000 | 4.000000 | 4.000000 | 2.000000 | 25274.000000 |
| max | 204887.000000 | 61.000000 | 36.000000 | 5.000000 | 6.000000 | 8.000000 | 3.000000 | 38677.000000 |

**After**

| | CustomerID | Age | NumberOfPersonVisiting | NumberOfFollowups | NumberOfChildrenVisiting | NumberOfTrips_std | MonthlyIncome_std | DurationOfPitch_std |
|---|---|---|---|---|---|---|---|---|
| count | 4787.000000 | 4787.000000 | 4787.000000 | 4787.000000 | 4787.000000 | 4.787000e+03 | 4.787000e+03 | 4.787000e+03 |
| mean | 202445.630457 | 37.396490 | 2.906204 | 3.710257 | 1.190934 | -1.956562e-15 | -3.988581e-16 | -1.974954e-16 |
| std | 1412.764584 | 9.066171 | 0.725755 | 0.998026 | 0.858025 | 1.000104e+00 | 1.000104e+00 | 1.000104e+00 |
| min | 200000.000000 | 18.000000 | 1.000000 | 1.000000 | 0.000000 | -1.232865e+00 | -1.507295e+00 | -1.289506e+00 |
| 25% | 201225.500000 | 31.000000 | 2.000000 | 3.000000 | 1.000000 | -6.664805e-01 | -6.000160e-01 | -7.905575e-01 |
| 50% | 202446.000000 | 36.000000 | 3.000000 | 4.000000 | 1.000000 | -1.000963e-01 | -2.372884e-01 | -2.916091e-01 |
| 75% | 203673.500000 | 43.000000 | 3.000000 | 4.000000 | 2.000000 | 4.662879e-01 | 3.865745e-01 | 4.568135e-01 |
| max | 204887.000000 | 61.000000 | 5.000000 | 6.000000 | 3.000000 | 2.731825e+00 | 3.126298e+00 | 2.577344e+00 |

# Stage 2 (Data Preprocessing)

## Feature Encoding

Dilakukan label encoding dan one hot encoding pada fitur-fitur kategorikal

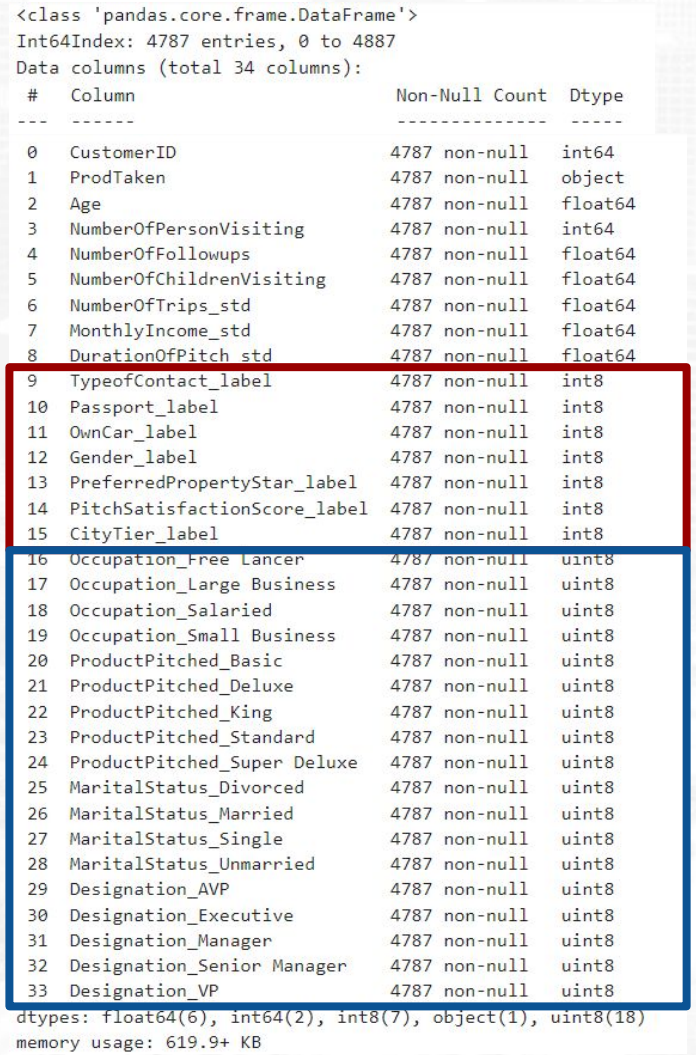
**Code:**

**Label Encoding → Fitur yang bertingkat/memiliki urutan**

```
df['TypeofContact_label'] = df['TypeofContact'].astype('category').cat.codes
df['Passport_label'] = df['Passport'].astype('category').cat.codes
df['OwnCar_label'] = df['OwnCar'].astype('category').cat.codes
df['Gender_label'] = df['Gender'].astype('category').cat.codes
df['PreferredPropertyStar_label'] = df['PreferredPropertyStar'].astype('category').cat.codes
df['PitchSatisfactionScore_label'] = df['PitchSatisfactionScore'].astype('category').cat.codes
df['CityTier_label'] = df['CityTier'].astype('category').cat.codes
```

**One Hot Encoding → Fitur yang tidak bertingkat/tidak memiliki urutan**

```
col = ['Occupation','ProductPitched','MaritalStatus','Designation']
for cat in col:
    onehots = pd.get_dummies(df[cat], prefix=cat)
    df = df.join(onehots)
```



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4787 entries, 0 to 4887
Data columns (total 34 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   CustomerID                    4787 non-null   int64
 1   ProdTaken                     4787 non-null   object
 2   Age                           4787 non-null   float64
 3   NumberOfPersonVisiting        4787 non-null   int64
 4   NumberOfFollowups             4787 non-null   float64
 5   NumberOfChildrenVisiting      4787 non-null   float64
 6   NumberOfTrips_std             4787 non-null   float64
 7   MonthlyIncome_std             4787 non-null   float64
 8   DurationOfPitch_std           4787 non-null   float64
 9   TypeofContact_label           4787 non-null   int8
 10  Passport_label                4787 non-null   int8
 11  OwnCar_label                  4787 non-null   int8
 12  Gender_label                  4787 non-null   int8
 13  PreferredPropertyStar_label   4787 non-null   int8
 14  PitchSatisfactionScore_label  4787 non-null   int8
 15  CityTier_label                4787 non-null   int8
 16  Occupation_Free Lancer        4787 non-null   uint8
 17  Occupation_Large Business     4787 non-null   uint8
 18  Occupation_Salaried           4787 non-null   uint8
 19  Occupation_Small Business     4787 non-null   uint8
 20  ProductPitched_Basic          4787 non-null   uint8
 21  ProductPitched_Deluxe         4787 non-null   uint8
 22  ProductPitched_King           4787 non-null   uint8
 23  ProductPitched_Standard       4787 non-null   uint8
 24  ProductPitched_Super Deluxe   4787 non-null   uint8
 25  MaritalStatus_Divorced        4787 non-null   uint8
 26  MaritalStatus_Married         4787 non-null   uint8
 27  MaritalStatus_Single          4787 non-null   uint8
 28  MaritalStatus_Unmarried       4787 non-null   uint8
 29  Designation_AVP               4787 non-null   uint8
 30  Designation_Executive         4787 non-null   uint8
 31  Designation_Manager           4787 non-null   uint8
 32  Designation_Senior Manager    4787 non-null   uint8
 33  Designation_VP                4787 non-null   uint8
dtypes: float64(6), int64(2), int8(7), object(1), uint8(18)
memory usage: 619.9+ KB
```

Label Encoding

One Hot Encoding

## Data Pre-Processing Conclusion

- Data awal : **4888 rows x 20 columns**
- Data setelah pre-processing (sampai encoding): **4787 rows x 34** columns (masih ada kolom CustomerID)

# Stage 3
Modeling

# Stage 3 (Modeling)

## Splitting Data Train and Test

**Code:**

```
from sklearn.model_selection import train_test_split
X = df[list(df.columns[2:])]
Y = df[['ProdTaken']]
X_train, X_test, y_train, y_test = train_test_split(X,Y,test_size = 0.2,random_state = 42)
df.shape
X_train.shape
X_test.shape
```

Memisahkan data train dan data test dengan ratio 80 : 20

| | Dataframe | Data Train | Data Test |
|---|---|---|---|
| Row | 4787 | 3829 | 958 |
| Column | 34 (include customerID) | 33 | 33 |

## Imbalanced Dataset (SMOTE)

**Code:**

```
y_train['ProdTaken'].value_counts()
```

Hasil tidak seimbang antara 0 dan 1

**Code:**

```
from imblearn import over_sampling
x_over_SMOTE, y_over_SMOTE =
over_sampling.SMOTE(0.5).fit_resample(x, y)
print('SMOTE')
print(pd.Series(y_over_SMOTE).value_counts())
```

Melakukan oversampling SMOTE dengan ratio 2 : 1

| Target | Before SMOTE 3829 x 33 | After SMOTE 4680 x 33 |
|---|---|---|
| 0 | 3120 | 3120 |
| 1 | 709 | 1560 |

# Stage 3 (ML Modeling & Evaluation)

| MODEL | MODEL PERFORMANCE | | | | CONFUSION METRICS | | | |
|---|---|---|---|---|---|---|---|---|
| | ACCURACY | PRECISION | RECALL | EXECUTION TIME | PREDICT T ACTUAL T | PREDICT F ACTUAL F | PREDICT T ACTUAL F | PREDICT F ACTUAL T |
| Logistic Regression | 0.84 | 0.81 | 0.31 | 0.043 s | 61 | 745 | 14 | 138 |
| Logistic Regression (hyperparameter tuning) | 0.84 | 0.82 | 0.30 | 0.290 s | 59 | 746 | 13 | 140 |
| Decision Tree | 0.90 | 0.78 | 0.71 | 0.021 s | 141 | 720 | 39 | 58 |
| Decision Tree Regularization | 0.83 | 0.74 | 0.27 | 0.363 s | 53 | 740 | 19 | 146 |
| Random Forest | 0.90 | 0.99 | 0.53 | 0.355 s | 105 | 758 | 1 | 94 |
| **XGBoost** | **0.93** | **0.95** | **0.70** | **0.219 s** | **140** | **752** | **7** | **59** |
| KNN | 0.87 | 0.88 | 0.41 | 0.004 s | 81 | 748 | 11 | 118 |
| KNN Regularization | 0.82 | 0.83 | 0.15 | 3.830 s | 29 | 753 | 6 | 170 |

- Jenis model prediksi yang dicoba terdiri dari : Logistic Regression, Logistic Regression (Hyperparameter Tuning), Decision Tree, Decision Tree Regularization, Random Forest, XGBoost, KNN, dan KNN Regularization.
- Jenis model prediksi yang dipakai adalah XGBoost dengan accuracy 0.93, precision 0.95, dan recall 0.70
- Model XGBoost dipilih karena memiliki precision dan recall yang tinggi. Kami mempertimbangkan recall dari model kami karena ingin mendapatkan customer yang convert sebanyak-banyaknya.

# Stage 4

## Business Metrics

# Stage 4 (Business Metrics)

## Assumption

### 1. Package Price

| Package | Price |
|---|---|
| Basic | **$1000** |
| Standard | **$2000** |
| Deluxe | **$3000** |
| Super Deluxe | **$4000** |
| King | **$5000** |

https://costaricaexperts.com/package/best/

### 2. Telemarketing for 1 customer = $50

https://www.magellan-solutions.com/blog/cost-of-telemarketing

# Stage 4 (Business Metrics)

## Calculation

Membagi customer berdasarkan paket travel yang dibeli

**Code:**

**Menjumlahkan customer yang convert dari masing-masing paket.**

```
Basic = Data_merge['ProductPitched_Basic'].sum()
Deluxe = Data_merge['ProductPitched_Deluxe'].sum()
King = Data_merge['ProductPitched_King'].sum()
Standard = Data_merge['ProductPitched_Standard'].sum()
SuperDeluxe = Data_merge['ProductPitched_Super Deluxe'].sum()
```

### Before

**Code:**

**Expected Revenue.**

```
ExpRev_before = (Basic*1000) + (Standard*2000) + (Deluxe*3000) +
(SuperDeluxe*4000) + (King*5000)
```

**Telemarketing Cost.**

```
Data_merge['TeleCost'] = ((Data_merge['DurationOfPitch'] * Data_merge
['NumberOfFollowups'])/60) * 50
TeleCost_before = Data_merge['TeleCost'].sum()
```

**Actual Revenue.**

```
ActRev_before = ExpRev_before*0.18
```

**Telemarketing cost.**

```
Spending_before = round((TeleCost_before/ActRev_before)*100,2)
```

### After

**Code:**

**Expected Revenue.**

```
ExpRev_after = (Basic*1000) + (Standard*2000) + (Deluxe*3000) +
(SuperDeluxe*4000) + (King*5000)
```

**Telemarketing Cost.**

```
Data_merge['TeleCost'] = ((Data_merge['DurationOfPitch'] * Data_merge
['NumberOfFollowups'])/60) * 50
TeleCost_after = Data_merge['TeleCost'].sum()
```

**Actual Revenue.**

```
ActRev_after = ExpRev_after*0.95
```

**Telemarketing cost.**

```
Spending_after = round((TeleCost_after/ActRev_after)*100,2)
```

# Stage 4 (Business Metrics)

## Potential Impact and Summary

| | **Before**<br>Data test<br>958 rows | **After**<br>Data test<br>958 rows<br>(assumption: after recall calculation) | |
|---|---|---|---|
| **Expected Revenue**<br>Sum(Price(i) * TotalCustomer(i)) | **$2,201,000** | **$2,201,000** | **Fixed** |
| **Telemarketing Cost**<br>DurationOfPitch*NumberOfFollowups*TeleCost | **$46,342.5** | **$46,342.5** | **Fixed** |
| **Actual Revenue**<br>Expected Revenue * BuyingPercentage | = $2,201,000*18%<br>**$396,180** | = $2,201,000*95%<br>**$2,090,950** | ▲ **427.8 %** |
| **Spending Revenue on Telemarketing Cost**<br>(Telemarketing cost / Actual Revenue)*100 | **11.7 %**** | **2.2 %** | ▼ **9.5 %** |

**Code:**

**Kenaikan Revenue.**

```
print(round(((ActRev_after-ActRev_before)/(ActRev_before))*100,1),'persen')
```

**Penurunan Spending Revenue on telemarketing Cost.**

```
print(round((Spending_before-Spending_after),1),'persen')
```

- **Sebelum** menggunakan model, expected revenue yang didapatkan jika semua orang mengambil paket travel sebesar 2,2 juta dolar dengan telemarketing cost sekitar 46000 dolar. Akan tetapi persentase yang mengambil paket hanya 18%, jadi actual revenue yang didapatkan hanya sebesar 396 ribu dolar, sehingga spending revenue untuk telemarketing sebesar 11.7%

- **Setelah** menggunakan model, dengan asumsi data test sudah dikalikan dengan recall, maka expected revenue dan telemarketing cost memiliki nilai yang sama dengan sebelum menggunakan model. Model ini dapat memprediksi 95% kemungkinan customer yang akan mengambil paket travel, sehingga actual revenue yang akan didapatkan sekitar 2 juta dolar dan spending revenue untuk telemarketing sebesar 2.2%.

# Stage 4 (Business Metrics)

## Potential Impact and Summary

| | **Before**<br>Data test<br>958 rows | **After**<br>Data test<br>958 rows<br>(assumption: after recall calculation) | |
|---|---|---|---|
| **Expected Revenue**<br>Sum(Price(i) * TotalCustomer(i)) | **$2,201,000** | **$2,201,000** | **Fixed** |
| **Telemarketing Cost**<br>DurationOfPitch*NumberOfFollowups*TeleCost | **$46,342.5** | **$46,342.5** | **Fixed** |
| **Actual Revenue**<br>Expected Revenue * BuyingPercentage | = $2,201,000*18%<br>**$396,180** | = $2,201,000*95%<br>**$2,090,950** | ▲ **427.8 %** |
| **Spending Revenue on Telemarketing Cost**<br>(Telemarketing cost / Actual Revenue)*100 | **11.7 %**** | **2.2 %** | ▼ **9.5 %** |

** **Warning!**
"You should spend 2–5% of your sales revenue on marketing"
https://nuphoriq.com/create-a-marketing-budget/

- Dengan actual revenue dan telemarketing cost yang sama, model ini dapat meningkatkan revenue sebesar 427.8% dan menurunkan spending revenue untuk telemarketing cost sebesar 9.5% dari nilai sebelum menggunakan model.
- Dari artikel yang kita temukan, kita hanya boleh mengeluarkan 2-5% dari actual revenue untuk keseluruhan kegiatan marketing, mencakup telemarketing, advertising, dll. Sehingga dapat disimpulkan bahwa pengeluaran biaya untuk telemarketing setelah menggunakan model sudah efisien.

# Final

## Business Recommendation

# Business Recommendation

Berdasarkan hasil EDA, tim kami menemukan jika setiap customer hanya ditawarkan produk tertentu berdasarkan designation yang mereka miliki (Executive - Basic, Manager - Deluxe, Senior Manager - Standard, AVP - Super Deluxe, VP - King). Namun, dengan cara seperti ini customer yang mengambil produk berdasarkan produk yang ditawarkan tidak banyak. Jenis produk yang paling diminati customer merupakan produk Basic dengan total 30% dari customer yang ditawarkan sementara produk yang lain tidak sampai 20%.

Sebaiknya, jika menawarkan produk pada customer tidak hanya menawarkan 1 produk tetapi semua produk yang ada karena berdasarkan data yang dimiliki saat ini, customer lebih tertarik dengan produk yang lebih murah.

# Terima Kasih