Rakamin Academy

Online Shoppers Purchasing Intention

Dokumen Laporan Final Project

oleh ec-Team (DS Batch 30 #9)

Latar Belakang Masalah

Peran ec-Team sebagai tim data scientist berperan dalam memimpin dan mengelola proyek analisis data untuk memprediksi niat beli pelanggan dengan tujuan membantu meningkatkan transaksi dan revenue perusahan e-commerce.

Tingginya minat belanja online telah menciptakan dimensi baru dalam bisnis toko online. Berdasarkan data 12,330 pengunjung situs toko online, hanya 15.5% atau sebanyak 1,910 pengunjung yang memutuskan untuk membeli produk dan melakukan transaksi, sedangkan 84.5% atau 10,422 pengunjung lainnya tidak melakukan transaksi. **Meskipun toko online memiliki jumlah pengunjung yang cukup banyak, persentase pengunjung yang melakukan transaksi masih relatif rendah. Hal ini dapat mengakibatkan potensi revenue tidak meningkat. Oleh karena itu, meningkatkan konversi penjualan dan memastikan pengunjung memutuskan untuk membeli produk dan menjadi pelanggan merupakan tantangan yang harus dihadapi oleh toko online.** Selain itu, perusahaan toko online juga memiliki peluang untuk mempelajari alasan pengunjung tidak membeli dan menemukan strategi yang tepat untuk meningkatkan pengalaman mereka di toko online agar lebih cenderung melakukan transaksi.

Goals, Objective, dan Business Metrics

***Goals* atau tujuan yang ingin dicapai oleh ec-Team dan perusahan toko online** adalah meningkatkan konversi penjualan dengan memahami pola dan niat beli pelanggan. Untuk mencapai tujuan bisnis tersebut, terdapat **dua objektif yang perlu dicapai**:

- Mengembangkan model prediktif untuk memprediksi niat beli pelanggan secara akurat dengan mengidentifikasi faktor-faktor yang paling mempengaruhi keputusan pembelian.
- Mengoptimalkan strategi penjualan untuk meningkatkan konversi penjualan dan kepuasan pelanggan.

Untuk mengukur keberhasilan pencapaian tujuan bisnis dan objektif yang telah ditetapkan, perlu adanya *business metrics*. ec-Team dan perusahaan toko online menetapkan ***Conversion Rate* (tingkat konversi) sebagai metrik bisnisnya** untuk memberikan gambaran seberapa efektif strategi penjualan yang telah dilakukan dalam meningkatkan konversi penjualan. Tingkat konversi dihitung dengan membagi jumlah visitor yang melakukan transaksi dengan jumlah total visitor, kemudian dikalikan dengan 100%.

Exploratory Data Analysis (EDA) – Descriptive Statistics

Rakamin Academy

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12330 entries, 0 to 12329
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Administrative           12330 non-null  int64
 1   Administrative_Duration  12330 non-null  float64
 2   Informational            12330 non-null  int64
 3   Informational_Duration   12330 non-null  float64
 4   ProductRelated           12330 non-null  int64
 5   ProductRelated_Duration  12330 non-null  float64
 6   BounceRates              12330 non-null  float64
 7   ExitRates                12330 non-null  float64
 8   PageValues               12330 non-null  float64
 9   SpecialDay               12330 non-null  float64
 10  Month                    12330 non-null  object
 11  OperatingSystems         12330 non-null  int64
 12  Browser                  12330 non-null  int64
 13  Region                   12330 non-null  int64
 14  TrafficType              12330 non-null  int64
 15  VisitorType              12330 non-null  object
 16  Weekend                  12330 non-null  bool
 17  Revenue                  12330 non-null  bool
dtypes: bool(2), float64(7), int64(7), object(2)
memory usage: 1.5+ MB
```

```
df[nums].describe().T
```

	count	mean	std	min	25%	50%	75%	max
Administrative	12330.0	2.315166	3.321784	0.0	0.000000	1.000000	4.000000	27.000000
Administrative_Duration	12330.0	80.818611	176.779107	0.0	0.000000	7.500000	93.256250	3398.750000
Informational	12330.0	0.503569	1.270156	0.0	0.000000	0.000000	0.000000	24.000000
Informational_Duration	12330.0	34.472398	140.749294	0.0	0.000000	0.000000	0.000000	2549.375000
ProductRelated	12330.0	31.731468	44.475503	0.0	7.000000	18.000000	38.000000	705.000000
ProductRelated_Duration	12330.0	1194.746220	1913.669288	0.0	184.137500	598.936905	1464.157214	63973.522230
BounceRates	12330.0	0.022191	0.048488	0.0	0.000000	0.003112	0.016813	0.200000
ExitRates	12330.0	0.043073	0.048597	0.0	0.014286	0.025156	0.050000	0.200000
PageValues	12330.0	5.889258	18.568437	0.0	0.000000	0.000000	0.000000	361.763742

```
df[cats1].describe().T
```

	count	mean	std	min	25%	50%	75%	max
SpecialDay	12330.0	0.061427	0.198917	0.0	0.0	0.0	0.0	1.0
OperatingSystems	12330.0	2.124006	0.911325	1.0	2.0	2.0	3.0	8.0
Browser	12330.0	2.357097	1.717277	1.0	2.0	2.0	2.0	13.0
Region	12330.0	3.147364	2.401591	1.0	1.0	3.0	4.0	9.0
TrafficType	12330.0	4.069586	4.025169	1.0	2.0	2.0	4.0	20.0

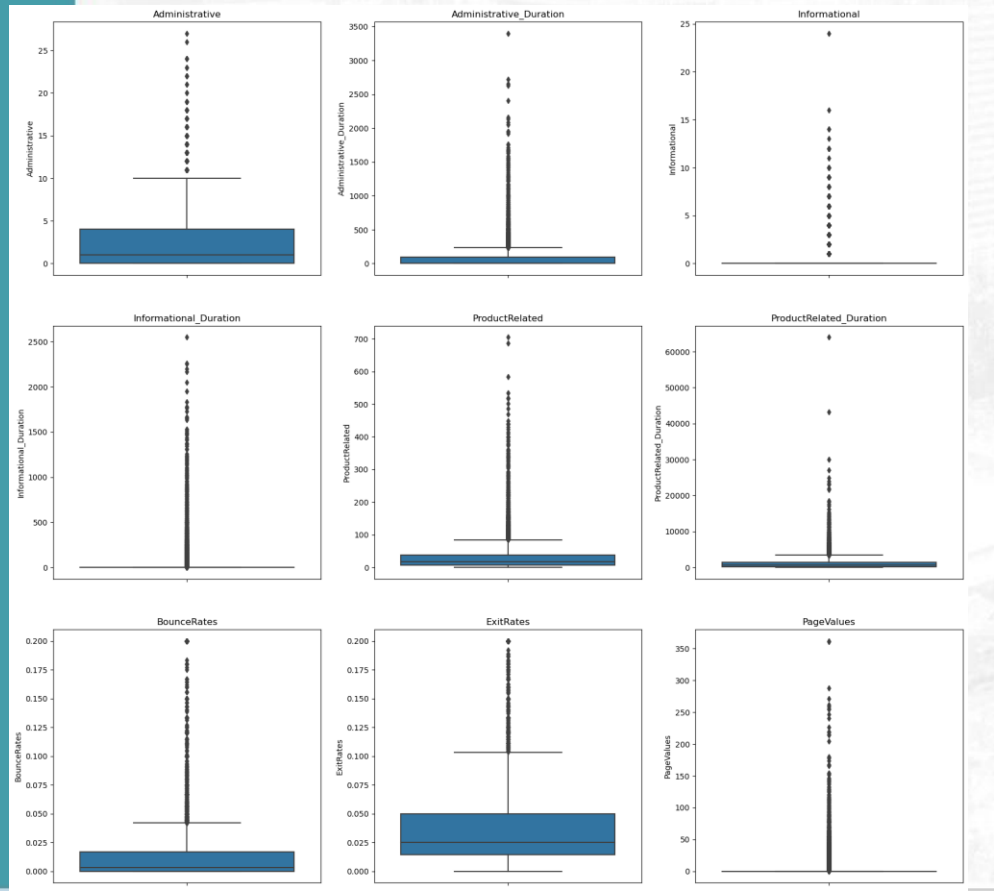
```
df[cats2].describe()
```

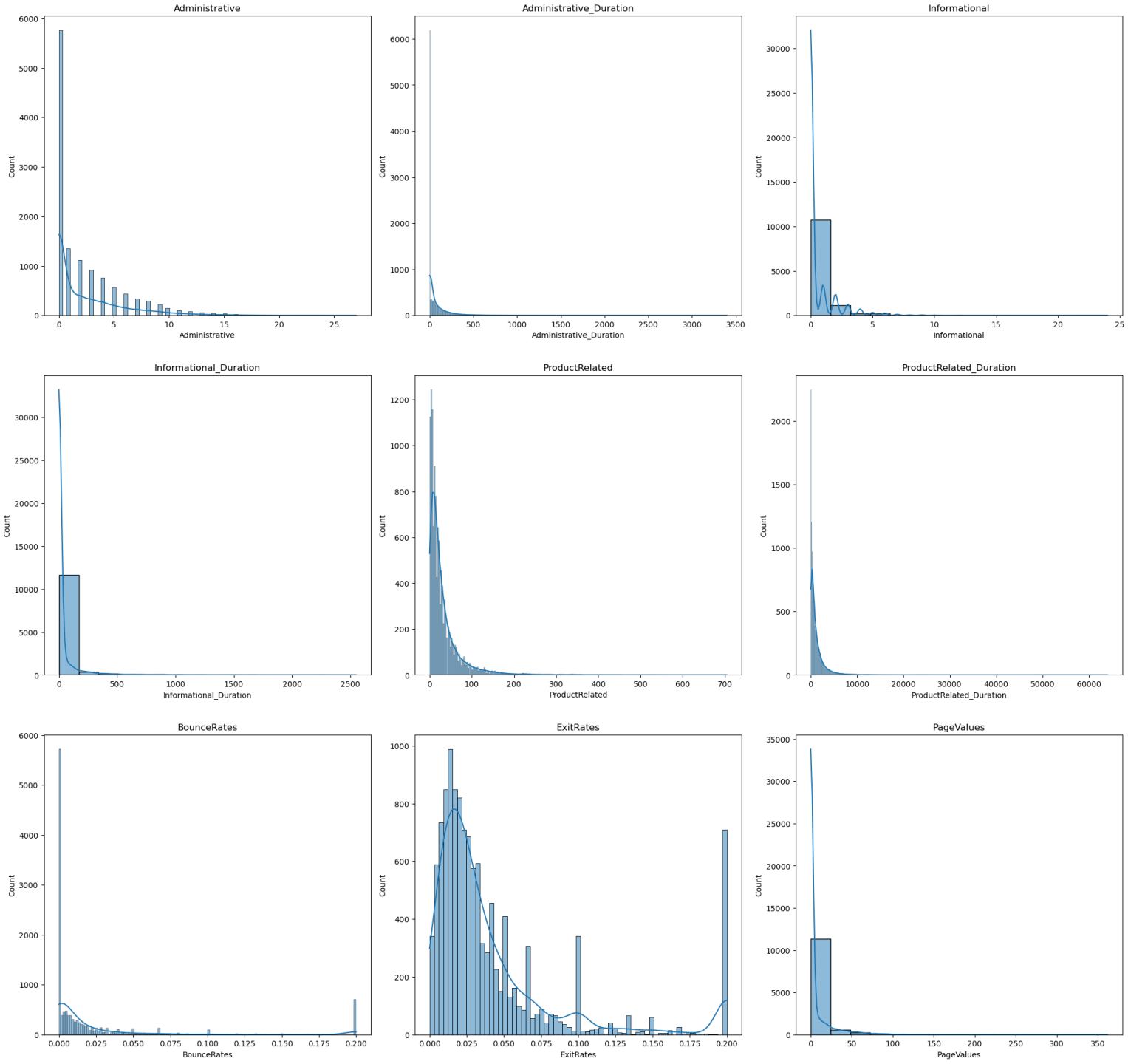
	Month	VisitorType	Weekend	Revenue
count	12330	12330	12330	12330
unique	10	3	2	2
top	May	Returning_Visitor	False	False
freq	3364	10551	9462	10422

- Dataset ini memiliki 12,330 row data dan 18 feature yang dibagi menjadi 9 feature numerical dan 9 feature categorical, dimana Revenue yang bersifat data kategorik merupakan feature target.
- Tipe data OperatingSystems, Browser, Region, dan TrafficType berupa integer yang tidak sesuai untuk data kategorik, seharusnya berupa string atau object.
- Nilai mean memiliki *gap* yang cukup jauh dengan nilai median, dikarenakan value pada feature numerik didominasi dengan nilai 0.

EDA – Univariate (Numerical)

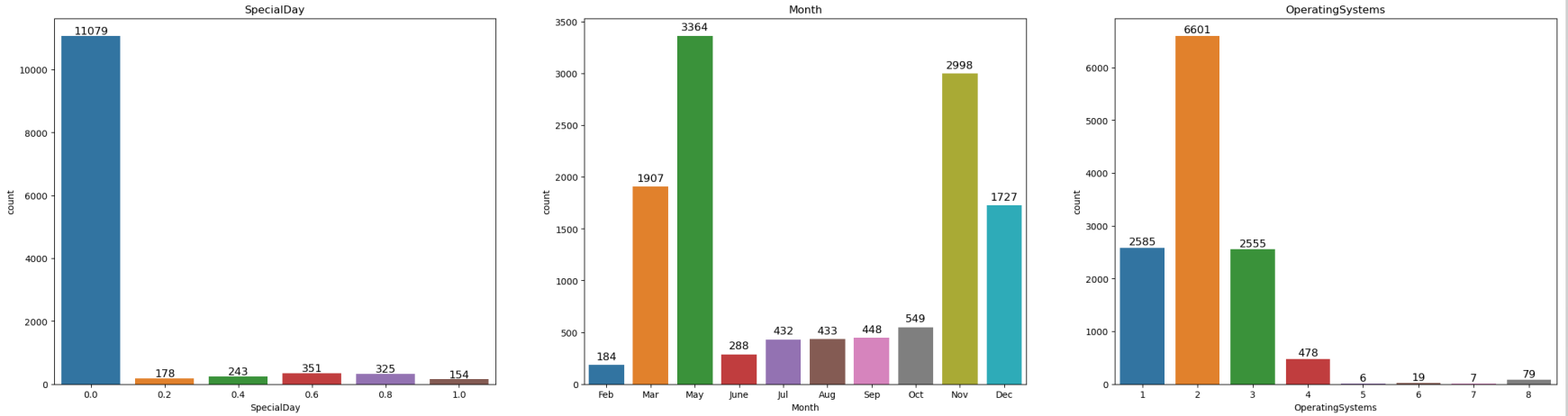
- Berdasarkan plot disamping, seluruh feature numerical memiliki distribusi yang skew ke kanan dengan mayoritas data bernilai 0 dan mendekati 0.
- Semua feature memiliki outlier yang cukup banyak jika dilihat dari boxplot dibawah.

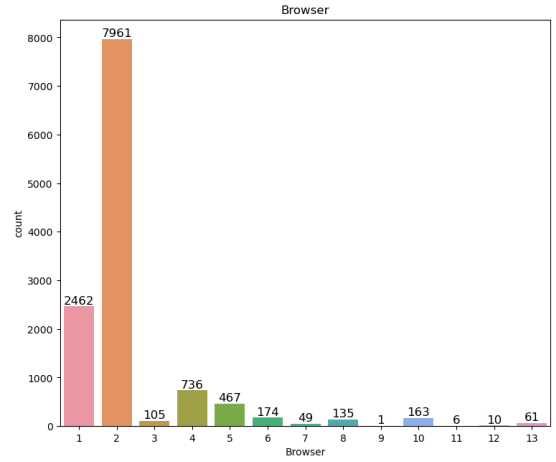


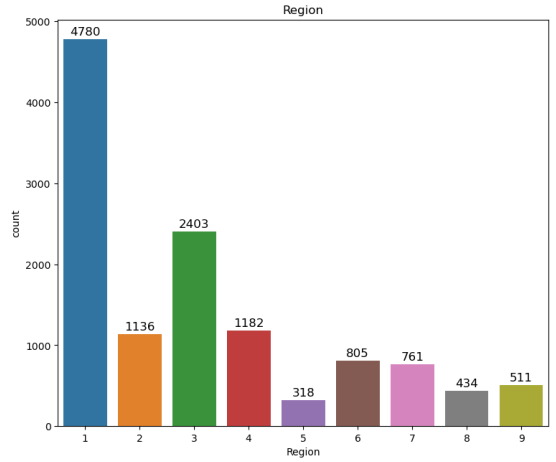


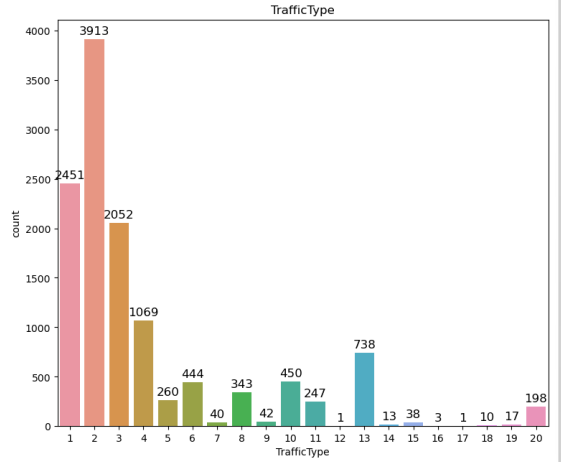
EDA – Univariate (Categorical)

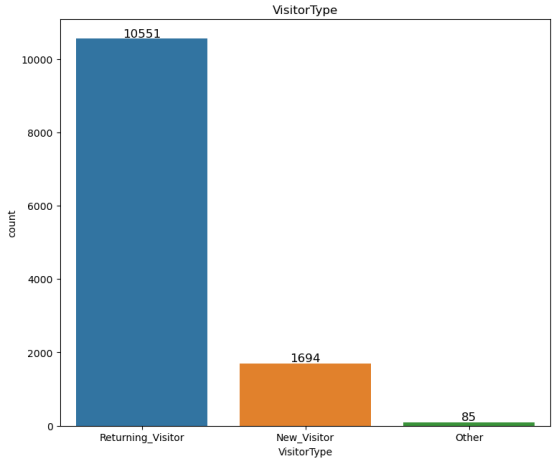
- Pada feature Month, tidak terekam data bulan Januari dan April.
- Pengunjung lebih dominan mengunjungi situs tidak dekat dengan Special Day dan tidak saat Weekend. Selain itu bulan Mei dan November memiliki pengunjung terbanyak.
- Operating System dan Browser 2 lebih dominan digunakan oleh pengunjung. Traffic Type 2 merupakan sumber yang paling banyak digunakan oleh pengunjung.
- Pengunjung terbanyak berasal dari Region 1 dan pengunjung lama (returning visitor).
- 10,422 (84.5%) pengunjung tidak melakukan transaksi, hanya 1908 (15.5%) yang menyumbang Revenue.

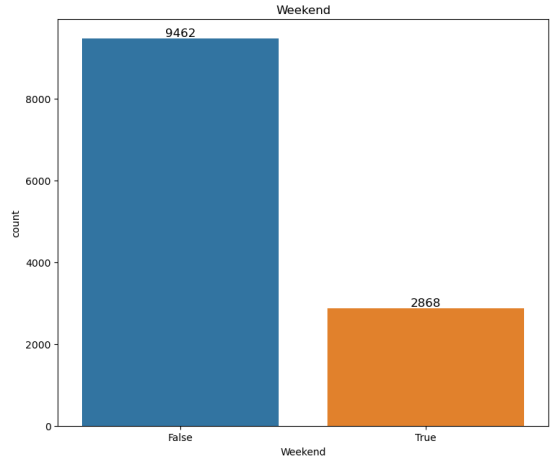


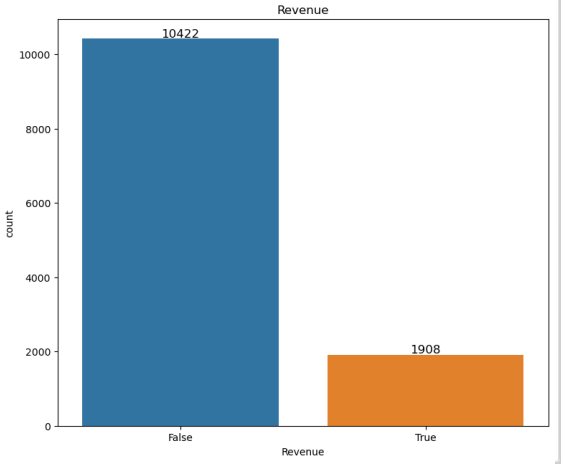






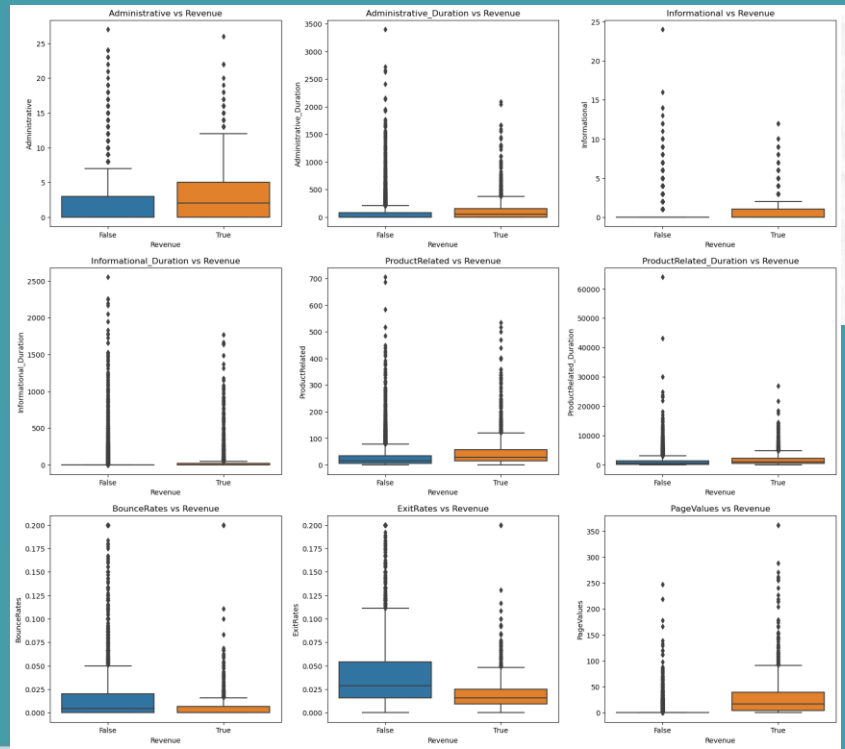


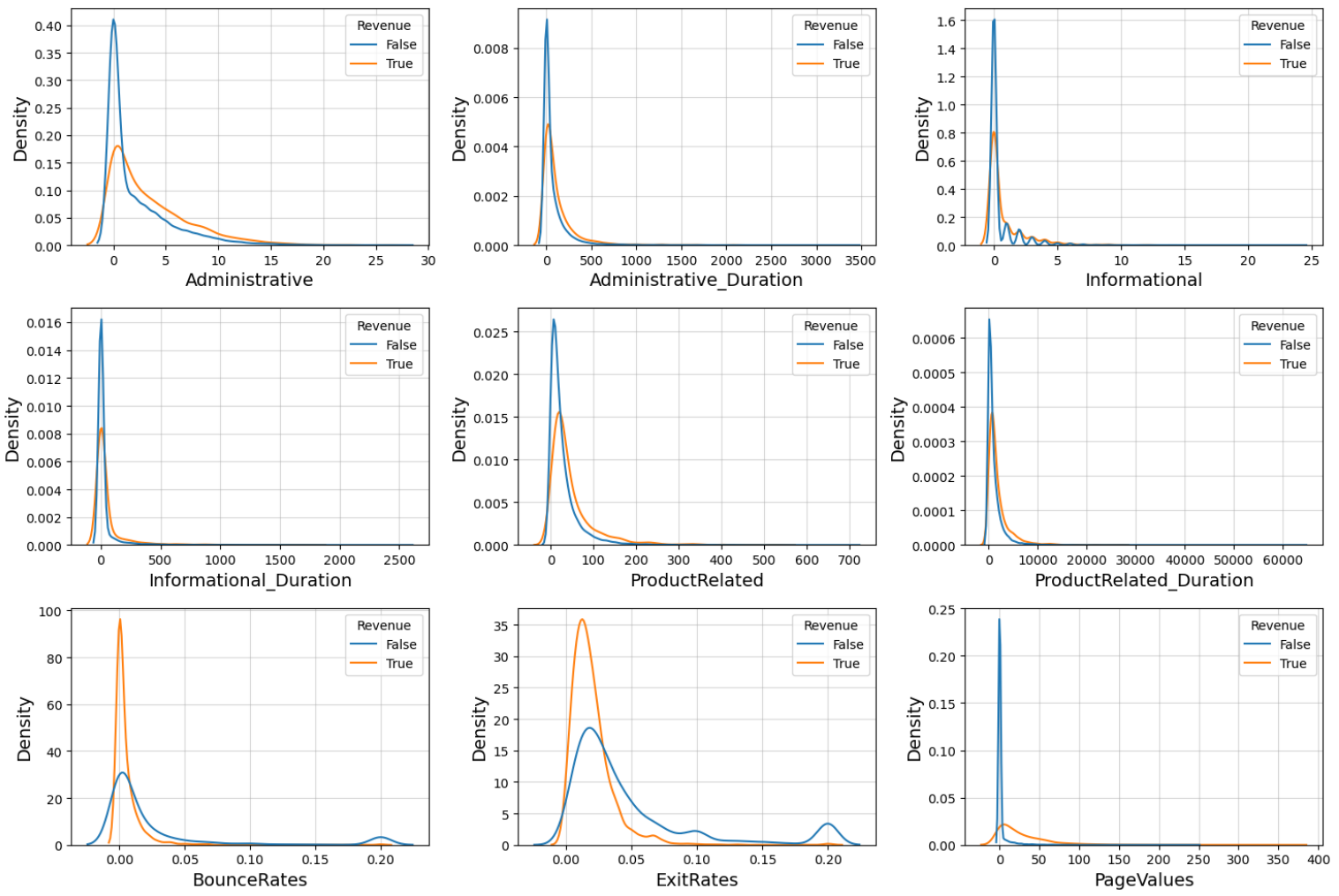




EDA – Multivariate (Numerical)

- Pengunjung yang mengunjungi halaman Produk Terkait secara intensif dan untuk waktu yang lama cenderung lebih mungkin untuk melakukan transaksi daripada pengunjung yang hanya mengakses halaman Administrative dan Informationalsaja.
- Jika pengunjung melakukan banyak pencarian produk, kemungkinan besar minat mereka untuk membeli akan meningkat dan mendorong mereka untuk melakukan transaksi.

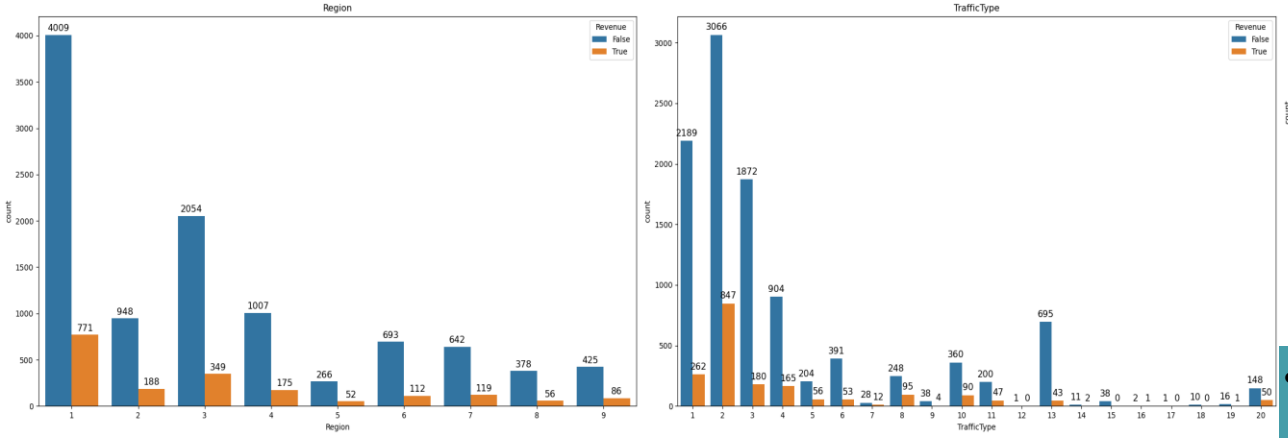


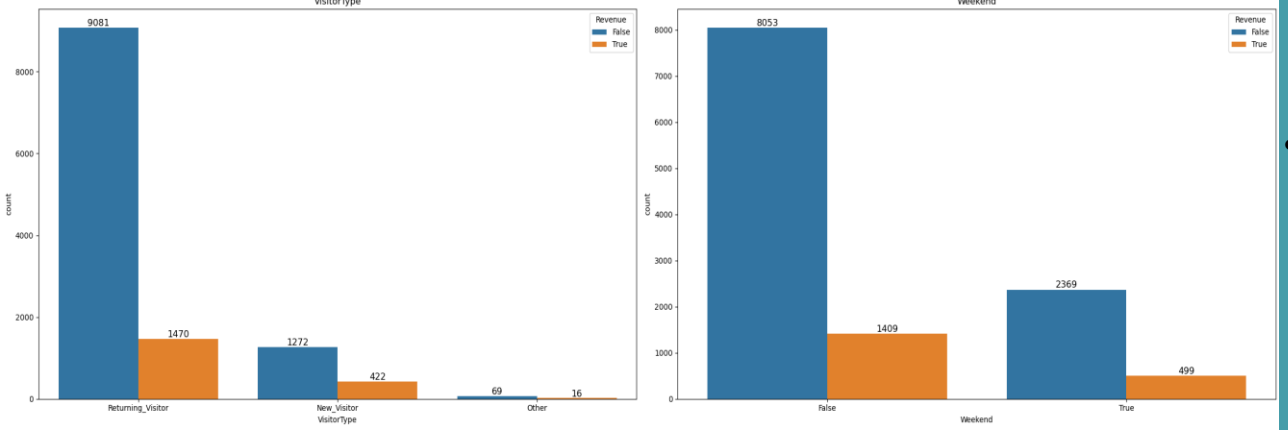


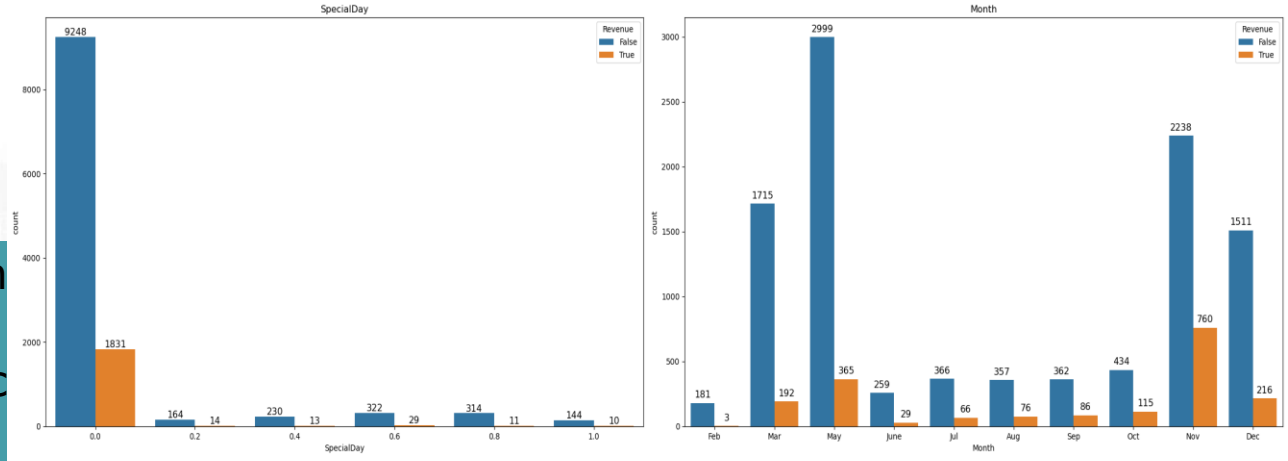
- Pengunjung dengan angka Bounce Rates dan Exit Rates yang tinggi cenderung tidak melakukan transaksi dan tidak menghasilkan revenue yang signifikan. Semakin tinggi tingkat Bounce dan Exit Rates, semakin besar kemungkinan bahwa pengunjung tidak tertarik dengan tawaran yang diberikan.
- Di sisi lain, pengunjung dengan angka Page Values yang tinggi lebih dominan melakukan transaksi. Hal tersebut dikarenakan semakin banyak pengunjung berinteraksi pada halaman keranjang dan checkout memiliki kemungkinan besar para pengunjung memutuskan untuk melakukan transaksi.

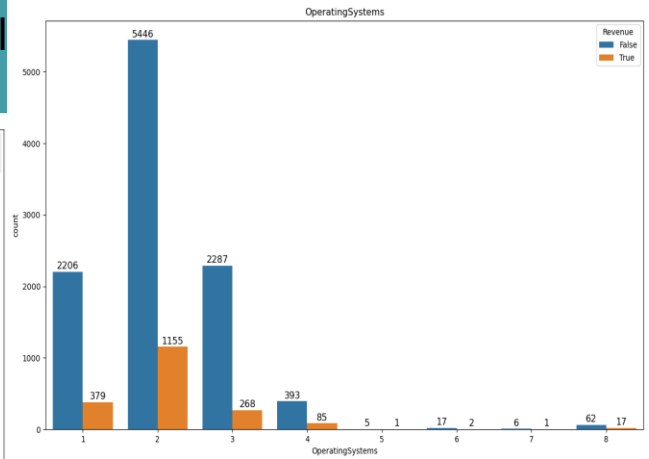
EDA – Multivariate (Categorical)

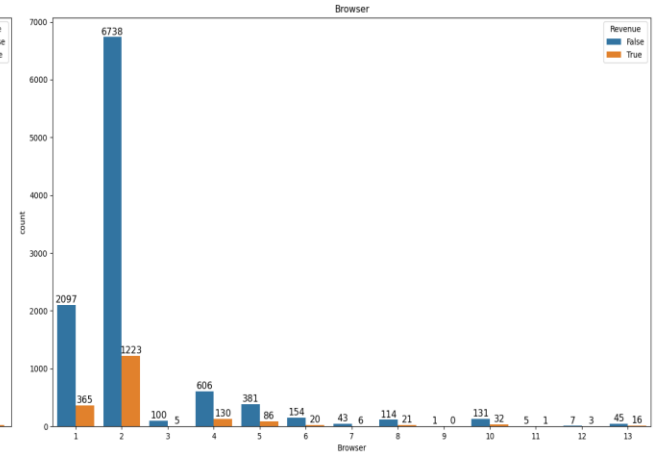
- Operating System dan Browser tipe 2 lebih banyak digunakan oleh pengunjung yang melakukan transaksi.
- Pengunjung yang melakukan transaksi bersumber dari Traffic Type 2 dan berasal dari Region 2.
- Pengunjung cenderung bertransaksi tidak dekat dengan Special Day dan tidak saat Weekend.







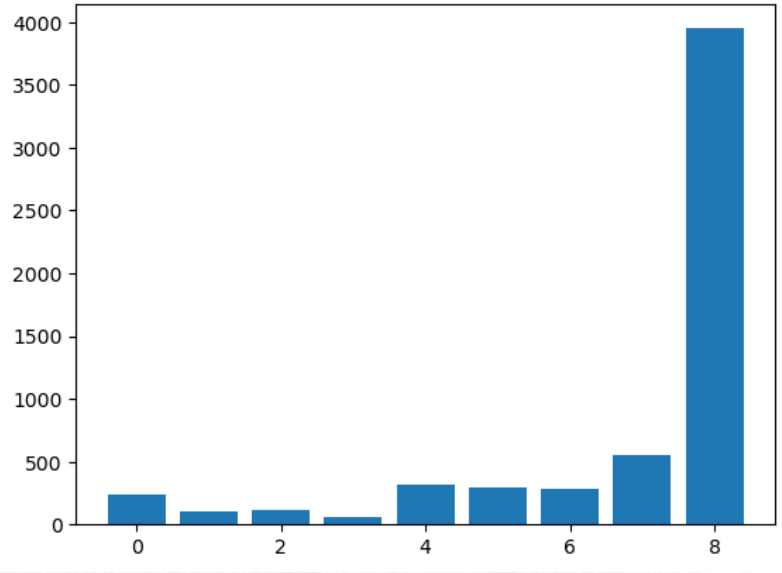




- Jumlah pengunjung terbanyak melakukan transaksi pada bulan November dan Desember, dikarenakan banyak event yang terjadi pada kedua bulan tersebut. Namun, bulan Mei memiliki lebih banyak pengunjung yang mengakses situs.
- Dilihat dari perbandingan antara pengunjung yang melakukan transaksi dengan yang tidak, tingkat konversi penjualan cenderung lebih tinggi pada New Visitor daripada tipe pengunjung yang lain. Walaupun begitu, Returning Visitor merupakan penyumbang revenue tertinggi jika dilihat dari jumlah pengunjung (count) yang melakukan transaksi.

EDA – Multivariate (Korelasi)

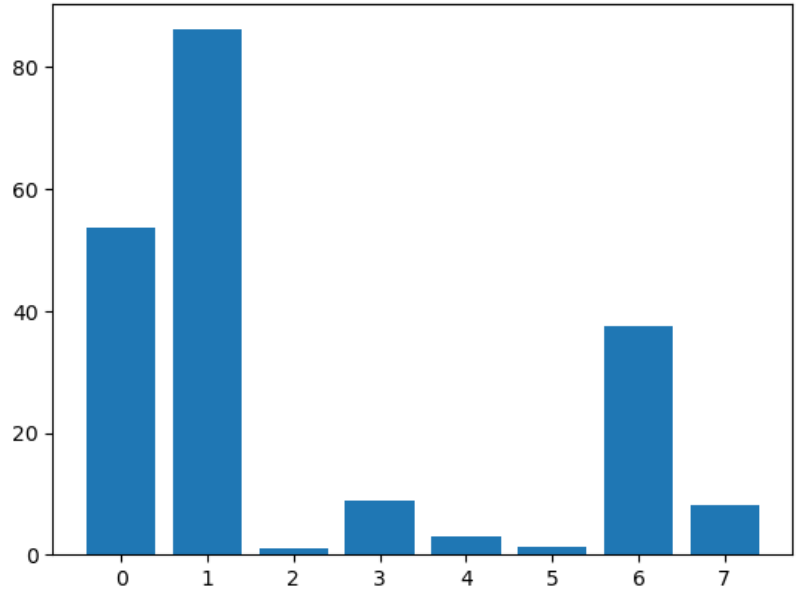
Feature Numerical & Target




```
Feature Administrative: 242.586667
Feature Administrative_Duration: 108.928515
Feature Informational: 112.751843
Feature Informational_Duration: 61.306613
Feature ProductRelated: 317.844350
Feature ProductRelated_Duration: 293.027603
Feature BounceRates: 286.375674
Feature ExitRates: 552.286502
Feature PageValues: 3949.262960
```

Berdasarkan hasil analisis korelasi feature numerical terhadap target dengan metode ANOVA, menunjukkan bahwa feature Page Values memiliki korelasi kuat dengan Revenue, diikuti oleh feature Exit Rates dan Product Related.

Feature Categorical & Target




```
Feature SpecialDay: 53.797094
Feature Month: 86.163696
Feature OperatingSystems: 1.037132
Feature Browser: 8.873291
Feature Region: 3.037565
Feature TrafficType: 1.283194
Feature VisitorType: 37.547523
Feature Weekend: 8.120464
```

Berdasarkan hasil analisis korelasi feature numerical terhadap target dengan metode Chi-square, menunjukkan bahwa feature Month memiliki korelasi kuat dengan Revenue, disusul oleh feature Special Day dan Visitor Type.

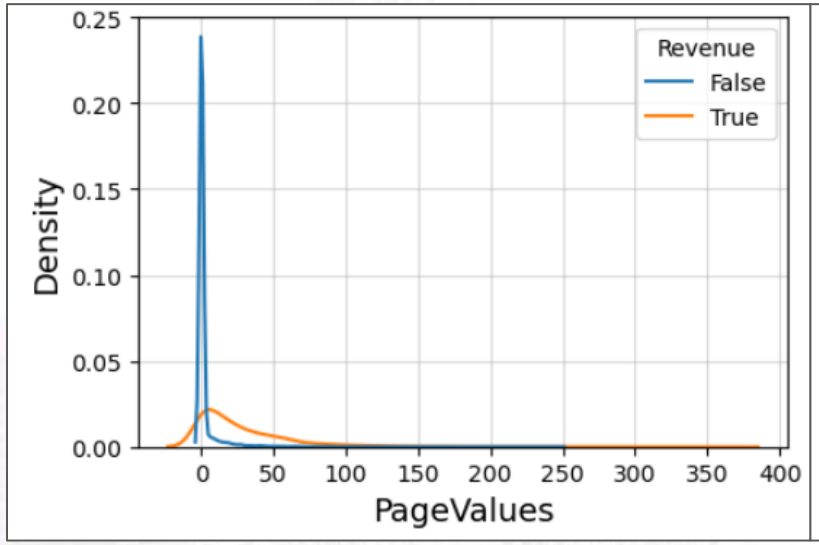
Business Insights

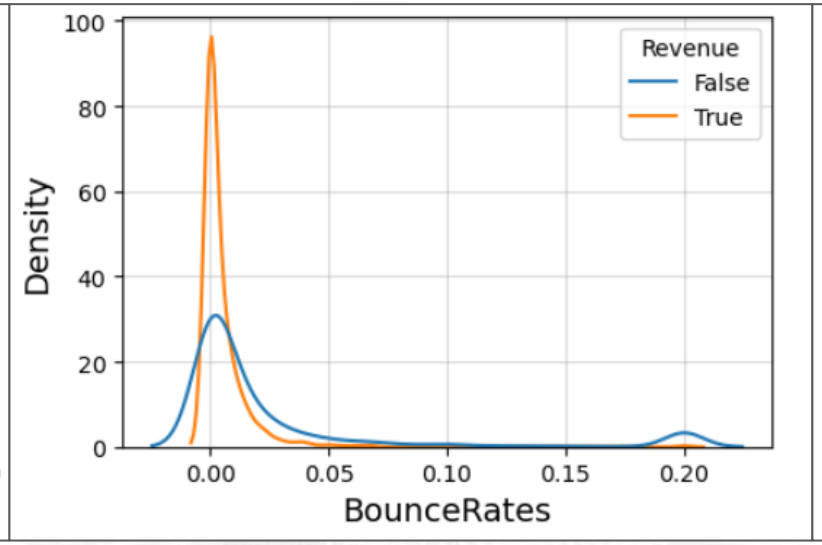
- Jumlah revenue atau pendapatan yang didapat dari pelanggan lama atau yang kembali lebih banyak daripada pelanggan baru. Namun, tingkat konversi pelanggan baru lebih tinggi dibandingkan dengan pelanggan lama. Dari total pengujung sebanyak 85% merupakan pengunjung kembali ke situs dan 15% pengunjung adalah baru. Kita dapat memberikan tawaran atau campaign untuk menarik lebih banyak pengunjung baru agar tertarik melakukan pembelian pada situs web dan membuat pelanggan lama untuk melakukan transaksi kembali di situs web.
- Sebanyak 65% pengunjung berasal dari browser 2 dan lebih dari 85% pengunjung berasal dari browser 1 dan 2. Kita dapat membuat situs web menjadi lebih menarik, interaktif, dan responsif terhadap browser ini. Selain itu, untuk meningkatkan konversi pada browser lainnya, kita dapat memasang iklan situs web pada browser lainnya.
- Wilayah 1 menyumbang penjualan lebih banyak diikuti oleh wilayah 3. Dengan informasi ini, dapat direncanakan campaign dan penyediaan pasokan barang dengan cara yang lebih baik. Sebagai contoh, kita mungkin mengusulkan untuk membangun gudang yang khusus melayani kebutuhan wilayah 1 untuk meningkatkan tingkat pengiriman dan memastikan bahwa produk dengan permintaan tertinggi selalu tersedia dengan baik.
- Pengunjung situs web tertinggi di bulan Mei, tetapi jumlah pembelian atau transaksi paling besar terjadi di bulan November. Hal ini perlu diselidiki lebih lanjut oleh tim bisnis untuk mengetahui apa yang menyebabkan atau faktor yang mempengaruhi tingginya transaksi pada bulan November.

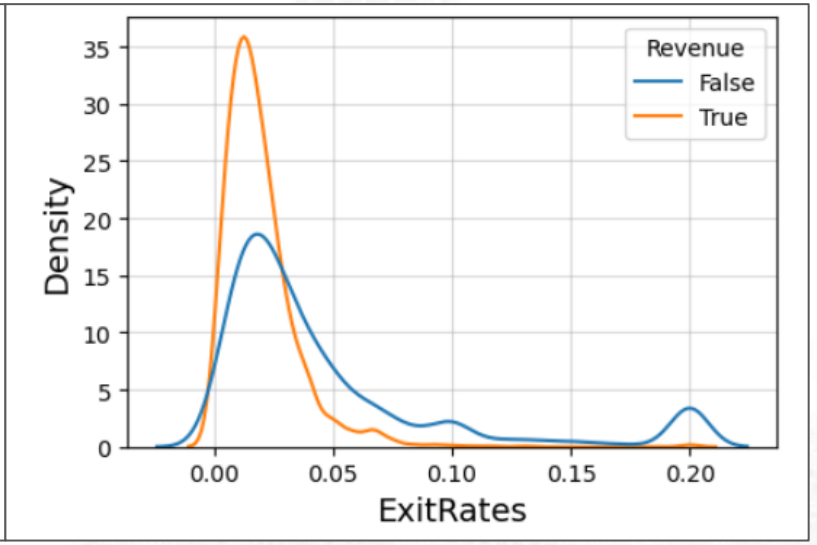
Business Insights

- Sekitar 95% pengunjung menggunakan operating system (OS) 1, 2, atau 3. Dengan mengetahui OS apa saja yang sering digunakan pelanggan untuk melakukan transaksi, bisa menjadi bahan pertimbangan jika kita ingin membuat aplikasi belanja yang user friendly. Dengan adanya aplikasi yang tersedia di aplikasi store di masing-masing OS dapat lebih memudahkan customer melakukan pencarian atau pembelian, serta memudahkan kita memberikan promosi dengan membuat notifikasi aplikasi.
- Konversi pengunjung pada hari weekdays lebih banyak yang tidak melakukan transaksi dibandingkan dengan hari weekend, namun jumlah pengunjung pada hari weekend masih terlalu rendah. Solusi yang akan kami lakukan adalah memprioritaskan pada hari weekend yang memiliki potensi konversi lebih tinggi dari hari weekdays dengan memberikan rekomendasi promosi diskon produk di hari weekend.

Insight - 1

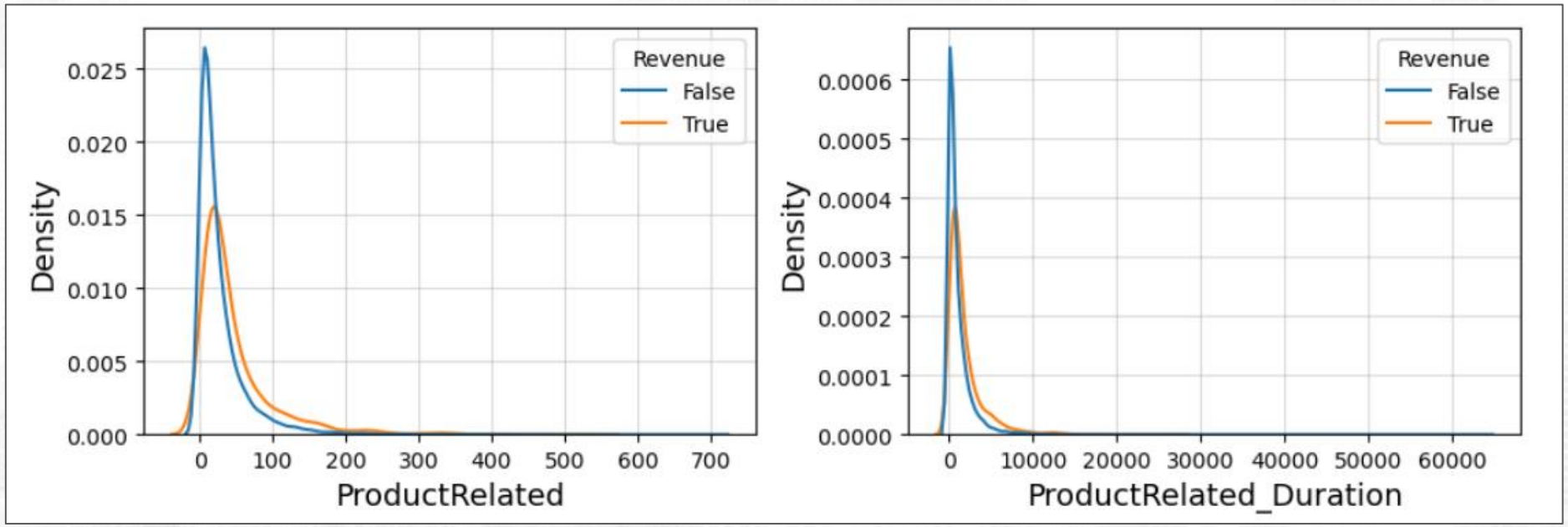






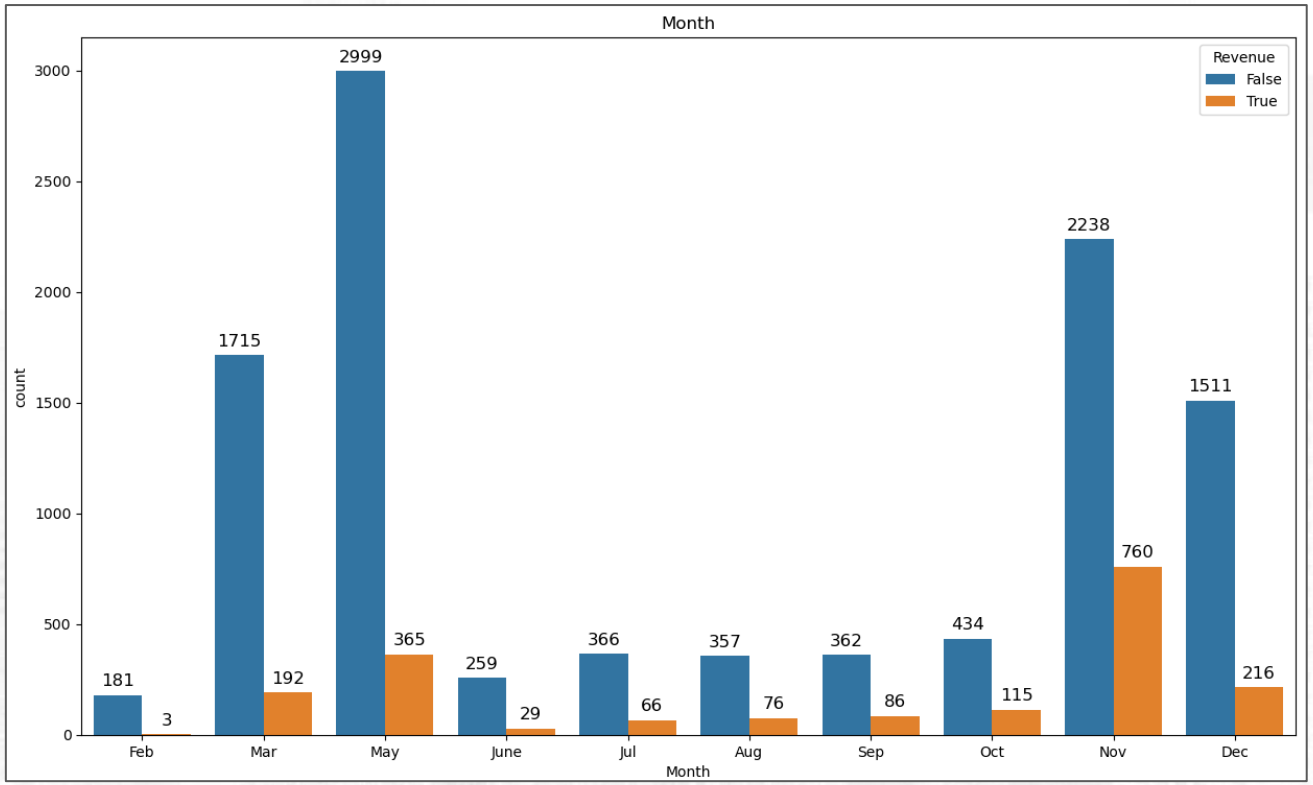
- Berdasarkan hasil analisis, **nilai Page Values yang tinggi cenderung ditemukan pada pengunjung yang melakukan transaksi**. Page Values dinilai dari seberapa banyak interaksi pengunjung pada halaman keranjang dan checkout yang berkemungkinan besar para pengunjung memutuskan untuk melakukan transaksi.
- Selain itu, dapat diketahui bahwa **lebih banyak pengunjung yang melakukan transaksi jika Bounce Rates dan Exit Rates rendah**. Bounce dan Exit Rates yang tinggi dapat mengindikasikan kurangnya minat pengunjung untuk berbelanja. Semakin tinggi tingkat Bounce dan Exit Rates, semakin besar kemungkinan bahwa pengunjung tidak tertarik dengan tawaran yang diberikan. Bounce Rates dinilai dari seberapa banyak pengunjung yang terjadi *bounce* atau hanya masuk di halaman awal kemudian keluar dari situs, sedangkan Exit Rates dinilai berdasarkan jumlah pengunjung yang sudah mengunjungi beberapa halaman pada situs namun memutuskan untuk keluar tanpa melakukan *action goals* (melakukan checkout atau memasukkan barang dalam keranjang).

Insight - 2



Berdasarkan hasil analisis, **pengunjung cenderung melakukan transaksi jika banyak mengunjungi halaman Product Related dan menghabiskan waktu pada halaman tersebut (Product Related Duration).** Pernyataan tersebut dapat diasumsikan jika pengunjung melakukan banyak pencarian produk, maka kemungkinan besar minat belinya akan meningkat dan mendorong mereka untuk melakukan transaksi.

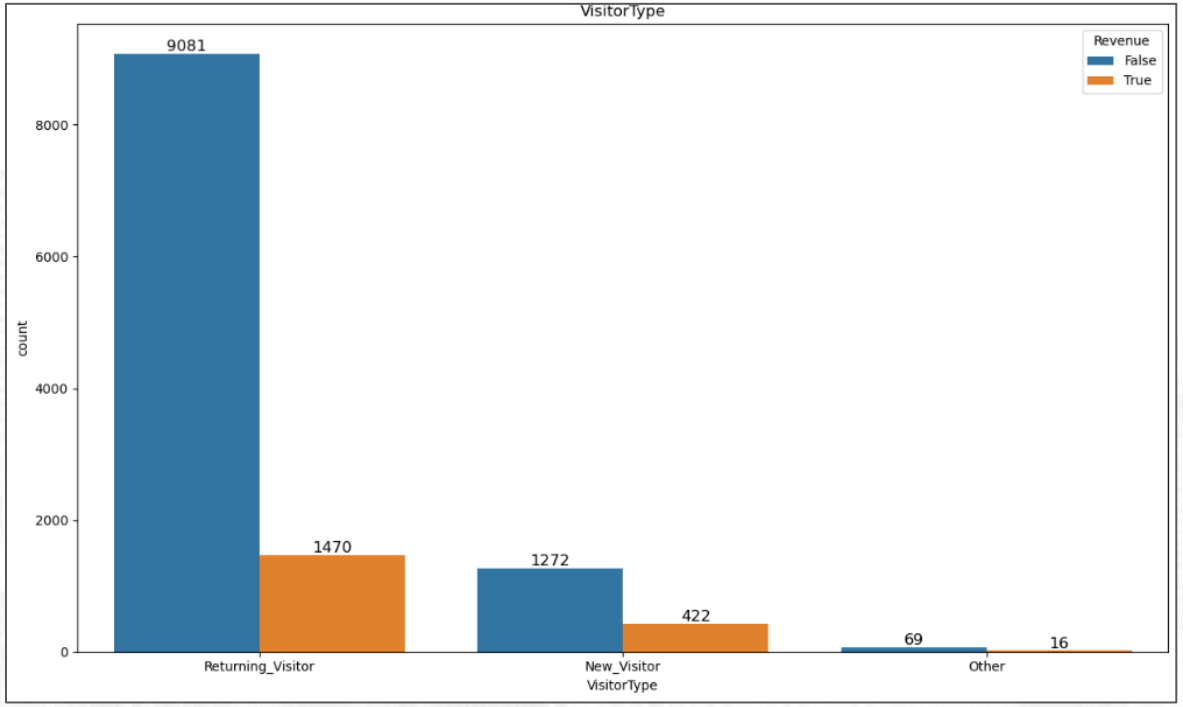
Insight – 3



Catatan: Data bulan Januari dan April tidak terekam

- **Pengunjung cenderung melakukan transaksi pada bulan November dan Desember, dikarenakan banyak event yang terjadi pada kedua bulan tersebut.** Namun, jumlah pengunjung tidak sebanyak pada bulan Mei, sehingga perlu adanya strategi untuk meningkatkan pengunjung situs toko online.
- **Bulan Mei merupakan bulan dengan pengunjung terbanyak** namun dominan dengan pengunjung yang tidak melakukan transaksi. Oleh karena itu, perlu adanya strategi penjualan yang dapat

Insight – 4



Pengunjung situs toko online dominan dengan Returning Visitor (pengunjung yang sebelumnya sudah pernah mengunjungi situs). Namun, jika dilihat dari perbandingan antara pengunjung yang melakukan transaksi dengan yang tidak, **tingkat konversi penjualan cenderung lebih tinggi pada New Visitor** daripada tipe pengunjung yang lain. Walaupun begitu, Returning Visitor merupakan penyumbang revenue tertinggi jika dilihat dari jumlah pengunjung (*count*) yang melakukan transaksi. Oleh karena itu, perlu dilakukan aksi untuk meningkatkan konversi penjualan terutama pada Returning Visitor.

Data Pre-Processing

1. Split Data

```python
# Memisahkan target variabel dari fitur
X = df_copy.drop("Revenue", axis=1)
y = df_copy["Revenue"]  # drop kolom 'Revenue' dari X

# Membagi data menjadi data pelatihan dan data pengujian
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)

# Menampilkan ukuran data pelatihan dan pengujian
print("Jumlah baris dan kolom data pelatihan:", X_train.shape)
print("Jumlah baris dan kolom data pengujian:", X_test.shape)
```

```
Jumlah baris dan kolom data pelatihan: (9864, 17)
Jumlah baris dan kolom data pengujian: (2466, 17)
```

- Split data dilakukan di awal pre-processing agar tidak terjadi data leakage.
- Data dibagi menjadi data train dan data test dengan rasio 80 : 20. Data Train nantinya akan digunakan untuk melatih model, sedangkan Data Test digunakan untuk menguji model yang sudah dilatih

Data Pre-Processing

2. Feature Extraction

Ada dua feature baru yang dapat dibuat dari feature-feature yang ada:

Nama Feature	Deskripsi Feature
Total_visit_duration	Jumlah durasi dari feature Administrative_Duration, Informational_Duration dan ProductRelated_Duration
Total_pageviews	Jumlah halaman yang dikunjungi dari feature Administrative, Informational, dan ProductRelated

```
df_copy["Total_visit_duration"] = (
    df_copy["Administrative_Duration"]
    + df_copy["Informational_Duration"]
    + df_copy["ProductRelated_Duration"]
)
df_copy["Total_visit_duration"] = df_copy["Total_visit_duration"].astype("int64")
df_copy["Total_visit_duration"].dtype
```

```
dtype('int64')
```

```
df_copy["Total_pageviews"] = (
    df_copy["Administrative"] + df_copy["Informational"] + df_copy["ProductRelated"]
)
df_copy["Total_pageviews"].dtype
```

```
dtype('int64')
```

```
df_copy[["Total_visit_duration", "Total_pageviews"]].describe().T
```

	count	mean	std	min	25%	50%	75%	max
Total_visit_duration	12330.0	1309.654015	2037.739593	0.0	222.0	680.0	1626.75	69921.0
Total_pageviews	12330.0	34.550203	46.514053	0.0	8.0	20.0	42.00	746.0

-Rekomendasi Feature Tambahan-

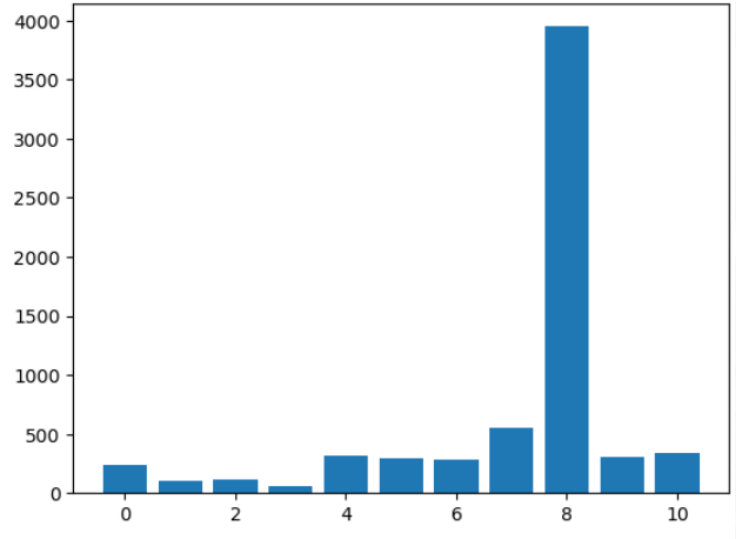
ec-Team memberikan beberapa rekomendasi fitur tambahan yang dapat meningkatkan performa model, yaitu:

- Penggunaan UserID / Invoice sebagai identifier
- Penambahan fitur Date-time untuk memprediksi waktu yang tepat dalam pengiriman campaign notification
- Penggunaan fitur Gender untuk mengoptimalkan rekomendasi produk
- Penggunaan Tanggal lahir (engan format DD/MM/YYYY) untuk optimalisasi dan campaign spesial diskon pada hari kelahiran
- Penggunaan Tanggal registrasi & Riwayat pembelian untuk menghitung customer lifetime value.

Data Pre-Processing

3. Feature Selection

```
Feature Administrative: 242.586667
Feature Administrative_Duration: 108.928515
Feature Informational: 112.751843
Feature Informational_Duration: 61.306613
Feature ProductRelated: 317.844350
Feature ProductRelated_Duration: 293.027603
Feature BounceRates: 286.375674
Feature ExitRates: 552.286502
Feature PageValues: 3949.262960
Feature Total_visit_duration: 307.726915
Feature Total_pageviews: 341.208963
```




```
# Model Built Using All Features

# prepare input data
def prepare_inputs(X_train, X_test):
    oe = OrdinalEncoder()
    oe.fit(X_train)
    X_train_enc = oe.transform(X_train)
    X_test_enc = oe.transform(X_test)
    return X_train_enc, X_test_enc


# prepare target
def prepare_targets(y_train, y_test):
    le = LabelEncoder()
    le.fit(y_train)
    y_train_enc = le.transform(y_train)
    y_test_enc = le.transform(y_test)
    return y_train_enc, y_test_enc


# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(
    Xcat, ycat, test_size=0.2, random_state=42
)
# prepare input data
X_train_enc, X_test_enc = prepare_inputs(X_train, X_test)
# prepare output data
y_train_enc, y_test_enc = prepare_targets(y_train, y_test)
# fit the model
model = LogisticRegression(solver="lbfgs")
model.fit(X_train_enc, y_train_enc)
# evaluate the model
yhat = model.predict(X_test_enc)
# evaluate predictions
accuracy = accuracy_score(y_test_enc, yhat)
print("Accuracy: %.2f" % (accuracy * 100))
```

```
Accuracy: 83.33
```

```
# define the evaluation method
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=42)
# define the pipeline to evaluate
model = LogisticRegression(solver="liblinear")
fs = SelectKBest(score_func=f_classif)
pipeline = Pipeline(steps=[("anova", fs), ("lr", model)])
# define the grid
grid = dict()
grid["anova__k"] = [i + 1 for i in range(Xnum.shape[1])]
# define the grid search
search = GridSearchCV(pipeline, grid, scoring="accuracy", n_jobs=-1, cv=cv)
# perform the search
results = search.fit(Xnum, ynum)
# summarize best
print("Best Mean Accuracy: %.3f" % results.best_score_)
print("Best Config: %s" % results.best_params_)
```

```
Best Mean Accuracy: 0.883
Best Config: {'anova__k': 2}
```

```
# Model Built Using Chi-Squared Features

# prepare input data
def prepare_inputs(X_train, X_test):
 oe = OrdinalEncoder()
 oe.fit(X_train)
 X_train_enc = oe.transform(X_train)
 X_test_enc = oe.transform(X_test)
 return X_train_enc, X_test_enc

# prepare target
def prepare_targets(y_train, y_test):
 le = LabelEncoder()
 le.fit(y_train)
 y_train_enc = le.transform(y_train)
 y_test_enc = le.transform(y_test)
 return y_train_enc, y_test_enc

# feature selection
def select_features(X_train, y_train, X_test):
 fs = SelectKBest(score_func=chi2, k=4)
 fs.fit(X_train, y_train)
 X_train_fs = fs.transform(X_train)
 X_test_fs = fs.transform(X_test)
 return X_train_fs, X_test_fs

# load the dataset
df = pd.read_csv('online_shoppers_intention.csv')
Xcat = df[['SpecialDay', 'Month', 'OperatingSystems', 'Browser', 'Region', 'TrafficType', 'VisitorType', 'Weekend']].values
ycat = df['Revenue'].values
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(Xcat, ycat, test_size=0.2, random_state=42)
# prepare input data
X_train_enc, X_test_enc = prepare_inputs(X_train, X_test)
# prepare output data
y_train_enc, y_test_enc = prepare_targets(y_train, y_test)
# feature selection
X_train_fs, X_test_fs = select_features(X_train_enc, y_train_enc, X_test_enc)
# fit the model
model = LogisticRegression(solver='lbfgs')
model.fit(X_train_fs, y_train_enc)
# evaluate the model
yhat = model.predict(X_test_fs)
# evaluate predictions
accuracy = accuracy_score(y_test_enc, yhat)
print('Accuracy: %.2f' % (accuracy*100))
```

```
Accuracy: 83.33
```

- Setelah menambahkan 2 fitur numerical yang baru dan diuji ANOVA kembali, 2 fitur tersebut score korelasinya masih lebih rendah dibandingkan PageValues, ExitRates dan ProductRelated.
- ec-Team sepakat untuk menggunakan seluruh fitur karena accuracy menggunakan seluruh fitur ataupun diseleksi dengan ANOVA dan Chi-Square menunjukkan nilai yang sama (83.33%). Selain itu, kami berasumsi dengan memotong banyak informasi / feature tidak menjamin performa model menjadi bagus.

Data Pre-Processing

Rakamin Academy

4. Handle Missing Values

Tidak perlu dilakukan handle missing values, karena dalam dataset ini tidak ada nilai null pada setiap featurenya.

```
df.isna().sum()
```

```
Administrative             0
Administrative_Duration    0
Informational              0
Informational_Duration     0
ProductRelated             0
ProductRelated_Duration    0
BounceRates                0
ExitRates                  0
PageValues                 0
SpecialDay                 0
Month                      0
OperatingSystems           0
Browser                    0
Region                     0
TrafficType                0
VisitorType                0
Weekend                    0
Revenue                    0
dtype: int64
```

5. Handle Duplicated Data

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

```
125
```

- Ditemukan 125 data duplikat.
- Data duplikat tidak didrop karena tidak ada feature identifier dan agar tidak mengganggu analisis dan pemodelan.

Data Pre-Processing

6. Handle Outliers

ec-Team mempertahakan outliers karena data didominasi dengan nilai 0, namun akan dihandle sekaligus pada saat transformasi untuk mengurangi jumlah outliersnya.

7. Feature Encoding

- Label encoding dilakukan untuk mengubah nilai fitur Revenue menjadi nilai numerik.
- Hot encoding dilakukan untuk mengubah nilai fitur VisitorType, Weekend, dan Month menjadi nilai numerik.

```
# One-hot encoding pada kolom Month, VisitorType, dan Weekenddf_copy.head(['SpecialDay','Month', 'OperatingSystems', 'Browser', 'Region', 'TrafficType','VisitorType', 'Weekend']).T

# Inisialisasi objek OneHotEncoder
encoder = OneHotEncoder()

# Pilih kolom-kolom yang akan di-encode
cat_cols = [
    "SpecialDay",
    "Month",
    "OperatingSystems",
    "Browser",
    "Region",
    "TrafficType",
    "VisitorType",
    "Weekend",
]

# Lakukan one-hot encoding pada kolom-kolom tersebut
cat_data_encoded = encoder.fit_transform(df_copy[cat_cols])

# Dapatkan daftar kategori dari kolom-kolom tersebut
cat_categories = encoder.categories_

# Gabungkan daftar kategori menjadi nama kolom baru
new_cols = [
    f"{col}_{val}" for col, vals in zip(cat_cols, cat_categories) for val in vals
]

# Konversi hasil encoding menjadi DataFrame
cat_data_encoded = pd.DataFrame(cat_data_encoded.toarray(), columns=new_cols)

# Label encoding pada kolom 'Revenue' dalam y
le = LabelEncoder()
y = le.fit_transform(y)

# Konversi kolom target 'Revenue' menjadi DataFrame
y = pd.DataFrame(y, columns=["Revenue"])

# Gabungkan cat_data_encoded dengan y
cat_data_encoded = pd.concat([cat_data_encoded, y], axis=1)
```

```
cat_data_encoded.head().T
```

	0	1	2	3	4
SpecialDay_0.0	1.0	0.0	1.0	1.0	1.0
SpecialDay_0.2	0.0	1.0	0.0	0.0	0.0
SpecialDay_0.4	0.0	0.0	0.0	0.0	0.0
SpecialDay_0.6	0.0	0.0	0.0	0.0	0.0
SpecialDay_0.8	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...
VisitorType_Other	0.0	0.0	0.0	0.0	0.0
VisitorType_Returning_Visitor	1.0	1.0	1.0	1.0	1.0
Weekend_False	1.0	1.0	1.0	1.0	1.0
Weekend_True	0.0	0.0	0.0	0.0	0.0
Revenue	0.0	0.0	0.0	0.0	0.0

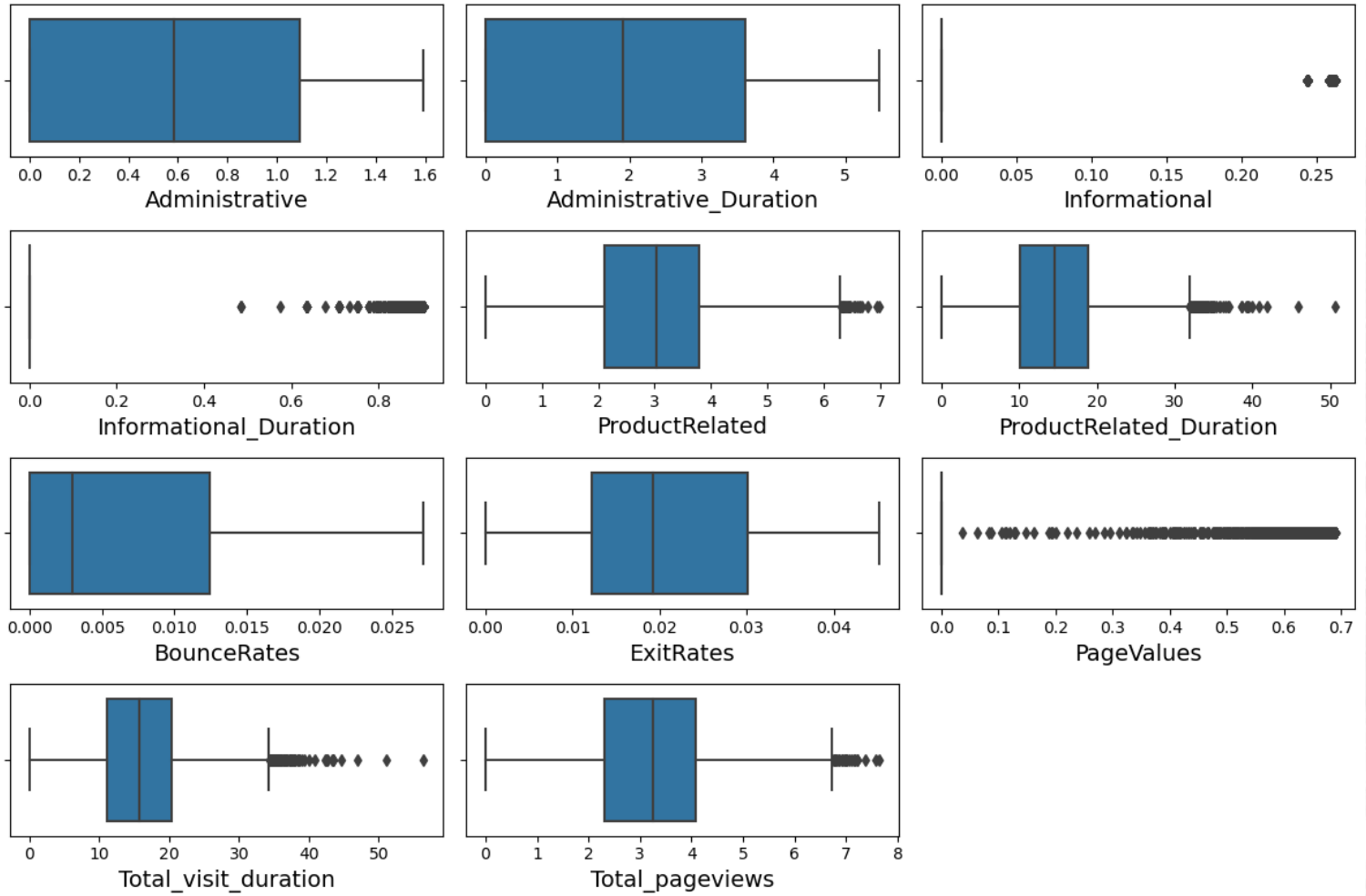
Data Pre-Processing

8. Feature Transformation

cc-Team menggunakan metode np.log1p(logx+1) dan Yeo-Johnson untuk mengubah distribusi data dari fitur numerik agar lebih mendekati distribusi normal sekaligus memangkas outliers. Berikut adalah tampilan akhir hasil distribusi fitur numerik setelah mengalami 2 transformasi tersebut:

```
plt.figure(figsize=(12, 8))
for i, col in enumerate(num_data):
    plt.subplot(4, 3, i + 1)
    sns.boxplot(x=np.log1p(num_data[col]), orient="v")
    plt.xlabel(col, fontsize=14)
    plt.tight_layout()

plt.show()

# save the figure to a file
fig.savefig("BOX POT OUTLIER.png", dpi=300, bbox_inches="tight")
```

```
plt.figure(figsize=(12, 8))
for i, col in enumerate(num_data):
    plt.subplot(4, 3, i + 1)
    sns.boxplot(x=stats.yeojohnson(num_data[col])[0], orient="v")
    plt.xlabel(col, fontsize=14)
    plt.tight_layout()

plt.show()

# save the figure to a file
fig.savefig("Feature Transformation.png", dpi=300, bbox_inches="tight")
```



Terlihat bahwa outliers setiap feature menjadi lebih sedikit dan lebih seimbang

Data Pre-Processing

9. Feature Scaling

Ec-Team menggunakan metode StandardScaler untuk melakukan scaling dan menormalisasi data numerik guna meningkatkan performa model.

```
# Inisialisasi objek StandardScaler
scaler = StandardScaler()

# Transformasi kolom-kolom dalam daftar nums
num_data_scaled = scaler.fit_transform(num_data)

# Buat DataFrame baru untuk data yang sudah di-scaling
num_data_final = pd.DataFrame(num_data_scaled, columns=num_data.columns)
```

```
num_data_final.describe()
```

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	Total_visit_duration	Total_pageviews
count	1.233000e+04	1.233000e+04	12330.000000	1.233000e+04	1.233000e+04	1.233000e+04	1.233000e+04	1.233000e+04	1.233000e+04	1.233000e+04	1.233000e+04
mean	-2.766103e-17	-3.457629e-17	0.000000	-1.844069e-17	5.532206e-17	1.844069e-17	3.688137e-17	1.844069e-17	9.220344e-18	-2.766103e-17	-1.844069e-17
std	1.000041e+00	1.000041e+00	1.000041	1.000041e+00	1.000041e+00	1.000041e+00	1.000041e+00	1.000041e+00	1.000041e+00	1.000041e+00	1.000041e+00
min	-6.969930e-01	-4.571914e-01	-0.396478	-2.449305e-01	-7.134884e-01	-6.243475e-01	-4.576830e-01	-8.863706e-01	-3.171778e-01	-6.427255e-01	-7.428208e-01
25%	-6.969930e-01	-4.571914e-01	-0.396478	-2.449305e-01	-5.560920e-01	-5.281214e-01	-4.576830e-01	-5.923930e-01	-3.171778e-01	-5.337768e-01	-5.708228e-01
50%	-3.959377e-01	-4.147639e-01	-0.396478	-2.449305e-01	-3.087548e-01	-3.113566e-01	-3.934903e-01	-3.686913e-01	-3.171778e-01	-3.090088e-01	-3.128257e-01
75%	5.072280e-01	7.035981e-02	-0.396478	-2.449305e-01	1.409492e-01	1.407881e-01	-1.109348e-01	1.425510e-01	-3.171778e-01	1.556179e-01	1.601688e-01
max	7.431499e+00	1.876956e+01	18.499599	1.786868e+01	1.513858e+01	3.280678e+01	3.667189e+00	3.229316e+00	1.916634e+01	3.367169e+01	1.529599e+01

10. Handle Class Imbalance

Untuk mengatasi ketidakseimbangan data yang terjadi, ec-Team menerapkan metode Class Weight pada saat pemodelan. Metode ini memberikan bobot yang lebih besar pada kelas yang jumlah datanya lebih sedikit, sehingga dapat membantu model dalam mempelajari pola pada kelas tersebut. Dengan penerapan metode Class Weight ini, diharapkan model dapat lebih efektif dalam melakukan prediksi pada kedua kelas meskipun jumlah data yang tidak seimbang.

Modelling

A. Split Data

```
# Memisahkan target variabel dari fitur
X = df_final.drop("Revenue", axis=1)
y = df_final["Revenue"]

# Membagi data menjadi data pelatihan dan data pengujian
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)

# Menampilkan ukuran data pelatihan dan pengujian
print("Jumlah baris dan kolom data pelatihan:", X_train.shape)
print("Jumlah baris dan kolom data pengujian:", X_test.shape)
```

```
Jumlah baris dan kolom data pelatihan: (9864, 82)
Jumlah baris dan kolom data pengujian: (2466, 82)
```

Untuk meningkatkan kualitas pemodelan, langkah awal yang perlu dilakukan adalah memisahkan data yang telah melalui preprocessing (df_final) menggunakan teknik stratified sampling menjadi 80% data train dan 20% data test. Hal ini bertujuan untuk mencegah terjadinya leakage.

B. Inisiasi Model yang digunakan

Berikut model klasifikasi yang umum digunakan untuk dataset yang memiliki banyak fitur dan kompleksitas yang cukup tinggi:

- Logistic Regression
- Decision Tree
- Random Forest
- kNN
- SVC
- Ada Boost
- Gradient BoostingClassifier
- XGBoost

```
# inisialisasi model
models = [
    [
        "Logistic Regression",
        LogisticRegression(class_weight="balanced", random_state=42),
    ],
    ["Decision Tree", DecisionTreeClassifier(class_weight="balanced", random_state=42)],
    ["Random Forest", RandomForestClassifier(class_weight="balanced", random_state=42)],
    ["KNeighborsClassifier", KNeighborsClassifier()],
    ["SVC", SVC(class_weight="balanced", probability=True, random_state=42)],
    ["AdaBoostClassifier", AdaBoostClassifier(random_state=42)],
    ["GradientBoostingClassifier", GradientBoostingClassifier(random_state=42)],
    ["XGBClassifier", XGBClassifier(random_state=42)],
]
```

Modelling

C. Model Evaluation – Cross Validation

	Model	Training Recall	CV Recall (mean)	CV Recall (std)	Training Precision	CV Precision (mean)	CV Precision (std)	Training F1	CV F1 (mean)	CV F1 (std)	Training AUC_ROC	CV AUC_ROC (mean)	CV AUC_ROC (std)
0	Logistic Regression	0.779295	0.764290	0.028131	0.525238	0.514997	0.012689	0.627501	0.615190	0.015626	0.911068	0.901156	0.007348
1	Decision Tree	1.000000	0.574202	0.013600	1.000000	0.564103	0.029835	1.000000	0.568768	0.018331	1.000000	0.745974	0.008790
2	Random Forest	1.000000	0.499338	0.023291	1.000000	0.768261	0.017989	1.000000	0.605137	0.021989	1.000000	0.923285	0.006857
3	KNeighborsClassifier	0.488930	0.386683	0.029846	0.799606	0.658836	0.005365	0.606756	0.486585	0.022133	0.939827	0.794322	0.017266
4	SVC	0.841469	0.759089	0.029813	0.608402	0.549736	0.011770	0.706176	0.637351	0.013412	0.945879	0.905363	0.004906
5	AdaBoostClassifier	0.596027	0.580052	0.036830	0.693412	0.678351	0.032958	0.641007	0.625207	0.034234	0.929315	0.915176	0.006177
6	GradientBoostingClassifier	0.662921	0.604135	0.031629	0.807306	0.727994	0.022422	0.727972	0.660198	0.027638	0.952645	0.931384	0.003362
7	XGBClassifier	0.944825	0.588515	0.028786	0.997768	0.701094	0.020305	0.970570	0.639737	0.024287	0.999244	0.924342	0.005392

- Metriks evaluasi yang cocok digunakan dataset ini adalah ROC-AUC score, karena ROC-AUC digunakan memprediksi kelas minoritas (True Revenue). Selain itu, ROC-AUC score dibutuhkan untuk melihat performa model setelah tuning threshold, sehingga asumsinya jika setelah dituning ROC-AUC score tinggi, maka nilai True Positive Rate (Recall) juga meningkat. Jika nilai True Positive Rate (target) meningkat maka performa model yang kita pilih itu sudah cukup baik dan dapat digunakan untuk model prediksi.

- Berdasarkan hasil cross validation, model **Gradient Boosting Classifier** memiliki nilai ROC-AUC score (test/mean) yang lebih tinggi dibandingkan model yang lain, dan cukup fit modelnya (gap antara score train dan test kecil).

Modelling

C. Model Evaluation – Cross Validation

	Model	Training Recall	CV Recall (mean)	CV Recall (std)	Training Precision	CV Precision (mean)	CV Precision (std)	Training F1	CV F1 (mean)	CV F1 (std)	Training AUC_ROC	CV AUC_ROC (mean)	CV AUC_ROC (std)
0	Logistic Regression	0.779295	0.764290	0.028131	0.525238	0.514997	0.012689	0.627501	0.615190	0.015626	0.911068	0.901156	0.007348
1	Decision Tree	1.000000	0.574202	0.013600	1.000000	0.564103	0.029835	1.000000	0.568768	0.018331	1.000000	0.745974	0.008790
2	Random Forest	1.000000	0.499338	0.023291	1.000000	0.768261	0.017989	1.000000	0.605137	0.021989	1.000000	0.923285	0.006857
3	KNeighborsClassifier	0.488930	0.386683	0.029846	0.799606	0.658836	0.005365	0.606756	0.486585	0.022133	0.939827	0.794322	0.017266
4	SVC	0.841469	0.759089	0.029813	0.608402	0.549736	0.011770	0.706176	0.637351	0.013412	0.945879	0.905363	0.004906
5	AdaBoostClassifier	0.596027	0.580052	0.036830	0.693412	0.678351	0.032958	0.641007	0.625207	0.034234	0.929315	0.915176	0.006177
6	GradientBoostingClassifier	0.662921	0.604135	0.031629	0.807306	0.727994	0.022422	0.727972	0.660198	0.027638	0.952645	0.931384	0.003362
7	XGBClassifier	0.944825	0.588515	0.028786	0.997768	0.701094	0.020305	0.970570	0.639737	0.024287	0.999244	0.924342	0.005392

- Metriks evaluasi yang cocok digunakan dataset ini adalah ROC-AUC score, karena ROC-AUC digunakan memprediksi kelas minoritas (True Revenue). Selain itu, ROC-AUC score dibutuhkan untuk melihat performa model setelah tuning threshold, sehingga asumsinya jika setelah dituning ROC-AUC score tinggi, maka nilai True Positive Rate (Recall) juga meningkat. Jika nilai True Positive Rate (target) meningkat maka performa model yang kita pilih itu sudah cukup baik dan dapat digunakan untuk model prediksi.

- Berdasarkan hasil cross validation, model **Gradient Boosting Classifier** memiliki nilai ROC-AUC score (test/mean) yang lebih tinggi dibandingkan model yang lain, dan cukup fit modelnya (gap antara score train dan test kecil).

Modelling

D. Hyperparameter Tuning

Sebelum tuning

	Model	Training Recall	CV Recall (mean)	CV Recall (std)	Training Precision	CV Precision (mean)	CV Precision (std)	Training F1	CV F1 (mean)	CV F1 (std)	Training AUC_ROC	CV AUC_ROC (mean)	CV AUC_ROC (std)
6	GradientBoostingClassifier	0.662921	0.604135	0.031629	0.807306	0.727994	0.022422	0.727972	0.660198	0.027638	0.952645	0.931384	0.003362

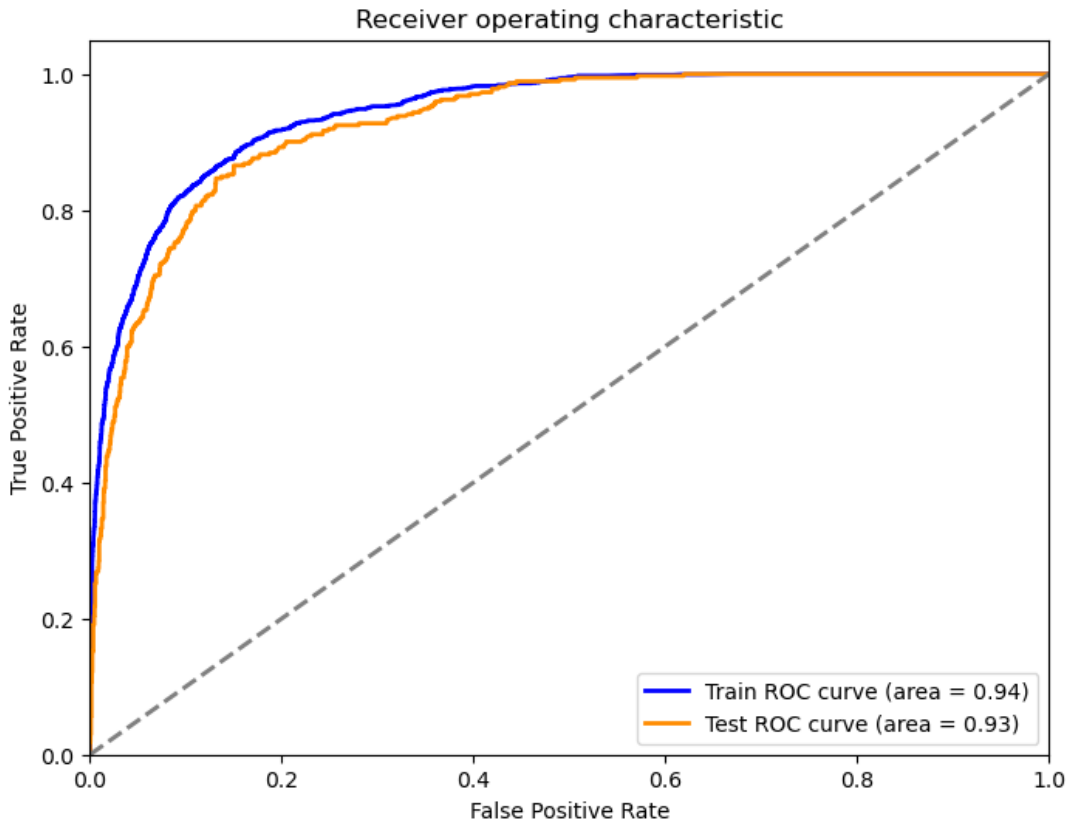
Setelah tuning

	model	best_params	Training Recall	CV Recall (mean)	CV Recall (std)	Training Precision	CV Precision (mean)	CV Precision (std)	Training F1	CV F1 (mean)	CV F1 (std)	Training AUC_ROC	CV AUC_ROC (mean)	CV AUC_ROC (std)
0	GradientBoostingClassifier	{'learning_rate': 0.05, 'max_features': None, ...	0.638669	0.609349	0.025337	0.783437	0.738455	0.019153	0.703586	0.667671	0.022895	0.94456	0.932141	0.003531

- Setelah dilakukan tuning, score ROC AUC (test/mean) meningkat dan gap antara train dan test menjadi model yang lebih fit dan lebik baik dari sebelumnya.
- ec-Team memutuskan tetap menggunakan model Gradient Boosting Classifier untuk model yang digunakan pada saat prediksi.

Modelling

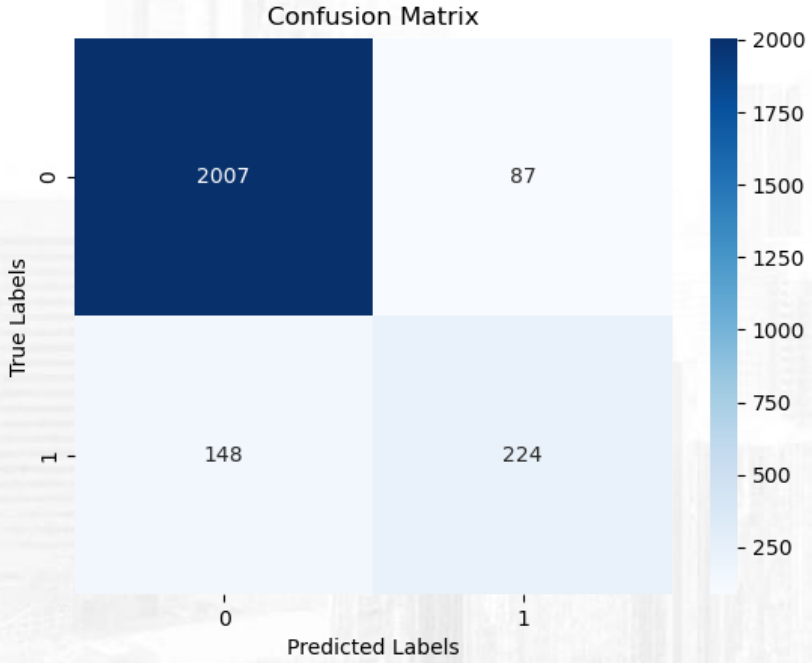
ROC AUC Curve



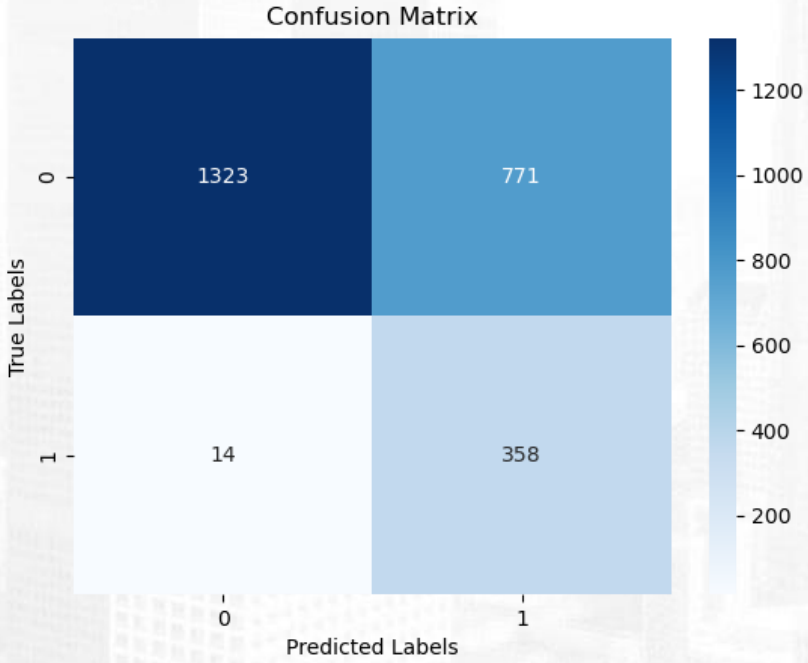
Berdasarkan grafik curve diatas, dapat dilihat bahwa dengan meningkatnya nilai ROC-AUC, maka true positive rate atau jumlah prediksi conversion rate (true Revenue) juga ikut meningkat.

Modelling

Confusion Matrix



Before Adjustment



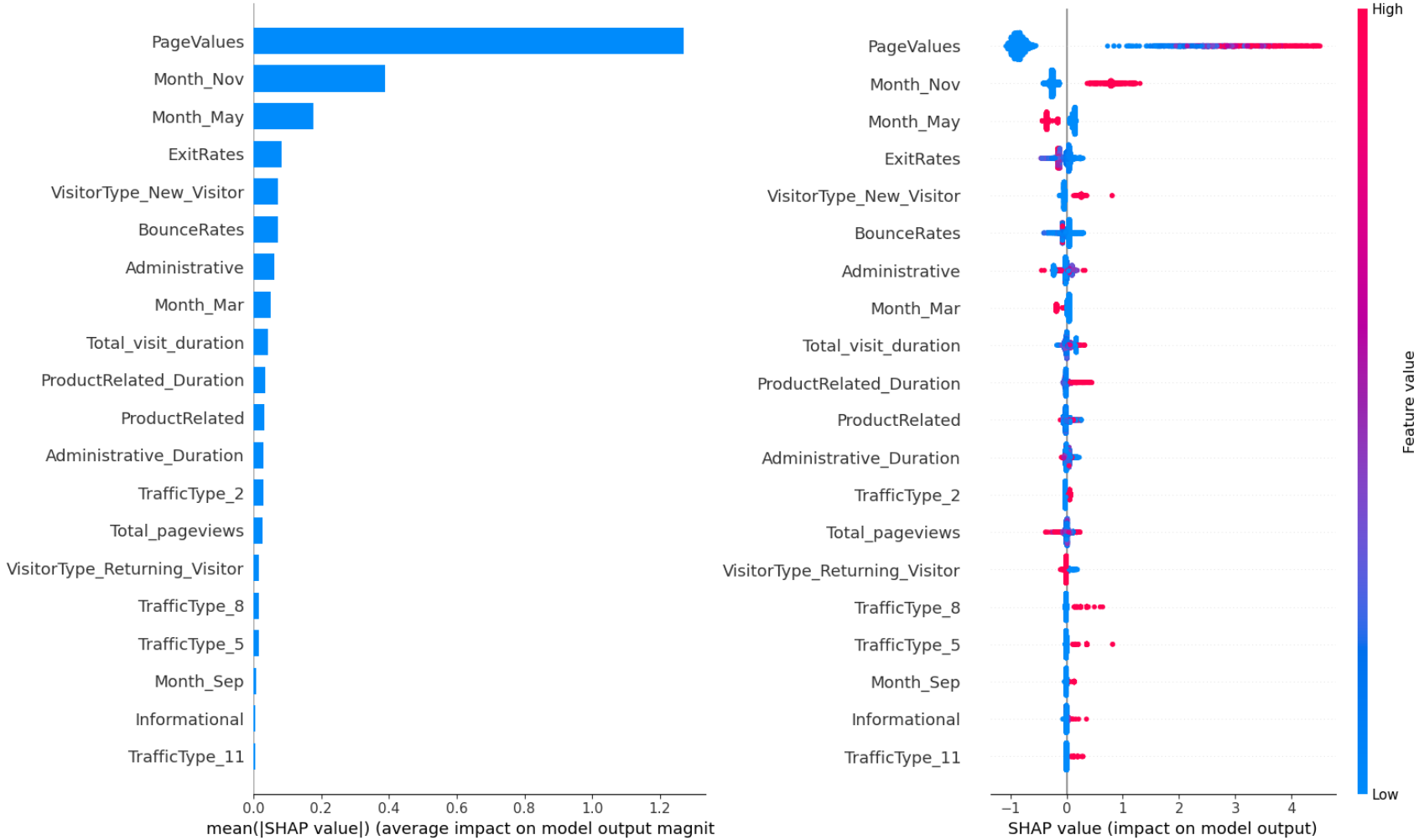
After Adjustment

Dapat dilihat bahwa, setelah dilakukan tuning, nilai false negative berkurang dari 148 menjadi 14 dan true positive meningkat dari 224 menjadi 358.

- False Negative (FN) terjadi ketika model memprediksi bahwa seseorang tidak akan membeli (negative), padahal kenyataannya orang tersebut membeli (positive).
- Dalam kasus prediksi apakah seseorang akan membeli atau tidak, FN sangat berbahaya karena dapat menyebabkan perusahaan kehilangan pelanggan potensial.
- Dalam kata lain, FN dapat menyebabkan perusahaan melewatkan peluang untuk menjual produknya pada orang yang sebenarnya berminat.
- Oleh karena itu, FN dapat dikurangi dengan meningkatkan recall sehingga pelanggan yang sebenarnya akan membeli produk dapat terdeteksi dengan lebih baik.

Modelling

E. Feature Importance



- Feature Page Values memiliki pengaruh paling besar terhadap hasil prediksi model.
- Feature Month Nov & May, Exit Rates dan Bounce Rates juga merupakan fitur penting yang cukup memiliki pengaruh yang signifikan.

Business Recommendation

Dalam meningkatkan konversi penjualan, ec-Team memberikan rekomendasi atau strategi bisnis untuk masing-masing feature yang memiliki dampak yang signifikan terhadap feature target, yaitu Revenue berdasarkan hasil analisis Feature Importance sebagai berikut:

Page Values

Mengoptimalkan Page Values dengan SEO atau atau optimasi mesin pencari dan memberikan voucher per-kategori produk. Dengan mengoptimalkan SEO, halaman website akan muncul lebih tinggi pada hasil pencarian Google, sehingga dapat menarik lebih banyak pengunjung dan meningkatkan kemungkinan terjadi konversi

Month

Memberikan promo dan diskon pada bulan November. Yang mana pada bulan November merupakan saat yang tepat untuk mempersiapkan diri untuk hari-hari libur dan perayaan. Selain itu, kami juga akan mengadakan event untuk menarik lebih banyak pengunjung dan memperkenalkan produk kami.

Business Recommendation

Bounce Rates

Menurunkan Bouces Rates dapat dilakukan dengan membuat tampilan situs web yang menarik dan mudah digunakan, menyediakan konten yang berkualitas, menyediakan navigasi yang jelas, mengoptimalkan kecepatan situs web, menyediakan tautan yang sesuai, dan meningkatkan keamanan situs web.

Exit Rates

Menurunkan exit rates dengan manambahkan Pop-up Exit Intent, CTA di tempat yang tepat, dan sediakan fitur live chat. Hal ini dapat memberikan pengalaman pengunjung yang lebih interaktif dan memudahkan mereka dalam menyelesaikan transaksi.

Kontribusi Anggota

Stage 0 & 1 dikerjakan bersama-sama

Mira Amelia Rosvita	Stage 2: Laporan Preprocessing & notulen mentoring, sedikit coding data cleaning Stage 3: Laporan Modelling, laporan notulen mentoring dan coding modelling Stage 4: Pembuatan script problem statement dan modeling, penyusunan PPT dan laporan & notulensi mentoring Stage 5: Pembuatan laporan final dan penyusunan notulen
Dania Dwi Pani	Stage 2: Laporan Preprocessing Stage 3: Kontribusi sangat kurang pada stage ini Stage 4: Penyusunan PPT Stage 5: Penyusunan PPT
Yanyan Gatot Mulyadi	Stage 2: Kontribusi besar dalam coding data cleaning hingga feature engineering Stage 3: Kontribusi besar dalam coding modelling Stage 4: Merapikan coding, pembuatan script confusion matrix dan potential revenue (simulasi model) Stage 5: Merapikan source code, pembuatan simulasi model dan perhitungan potential revenue
Haidar Aldi Eka Nugraha	Stage 2: Kontribusi kecil dalam coding data cleaning Stage 3: Coding modelling Stage 4: Penyusunan PPT Stage 5: Penyusunan PPT

Kontribusi Anggota

Stage 0 & 1 dikerjakan bersama-sama

Yanuar Wachyudi	Stage 2: Sedikit cleaning dan membuat git repository Stage 3: Coding modelling Stage 4: Pembuatan script pembuka, EDA & Preprocessing Stage 5: Merapikan interpretasi pada sourcecode
Muthmainah	Stage 2: Kontribusi besar dalam coding data cleaning hingga feature engineering Stage 3: Coding modelling Stage 4: Pembuatan script problem statement, preprocessing dan business recommendation, serta HW unsupervised learning Stage 5: Pembuatan laporan final dan rekomendasi bisnis
Muhamad Raihan Akbar	Stage 2: Laporan notulen mentoring Stage 3: Laporan notulen mentoring dan sedikit coding modelling Stage 4: Penyusunan PPT Stage 5: Penyusunan PPT & notulensi

Terima Kasih

Final Project Documents:
https://drive.google.com/drive/folders/1FpmY254TyeEVDhyJmi0HENSUSjDQejJ_

Git: *https://github.com/EC-Teams/Final-Project-Online-Shopping-Intention*