Rakamin Academy

E Commerce Purchasing Intention

Mau Beli Dokumen Final Project

(dipresentasikan setiap sesi mentoring)

Latar Belakang Masalah (1-2 slide)

Problem Statement : Pada sebuah website online shop, ada banyak returning visitor tetapi tingkat conversion rate dari returning visitor lebih rendah dibandingkan conversion rate new visitor.

Goal :

- Meningkatkan revenue (conversion rate) khususnya returning visitor dengan memfollow up user yang diprediksi akan melakukan pembelian/transaksi dengan Call To Action dan pemberian promo kepada target.

Objective :

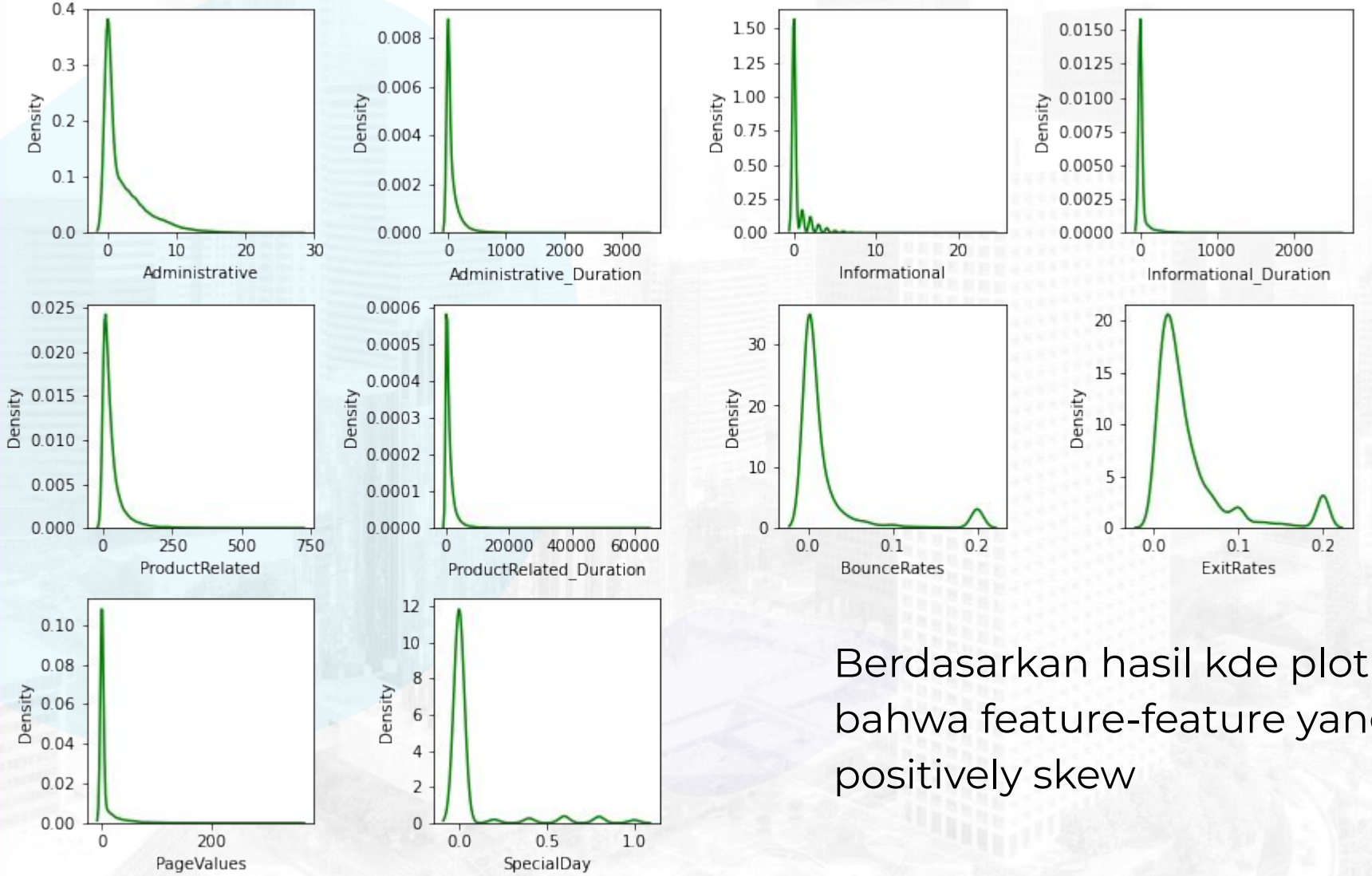
- Membuat predictive Machine Learning yang mampu memprediksi user yang mengunjungi website mempunyai intensi membeli atau tidak.

Business Metrics :

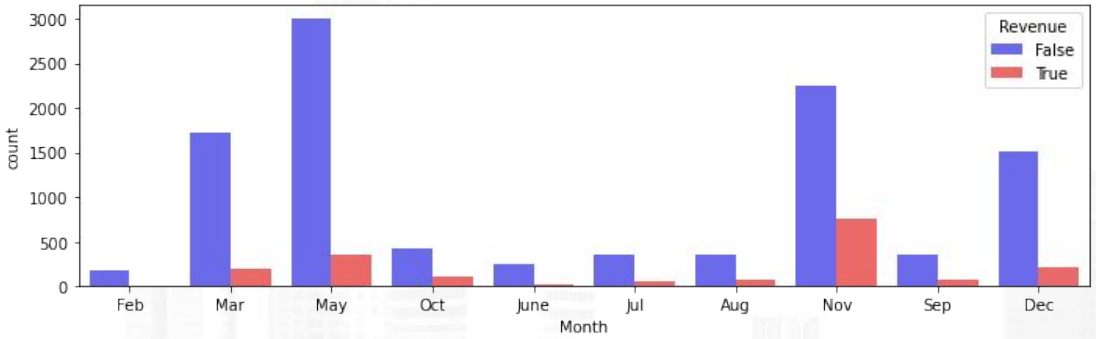
- Conversion Rate (tingkat konversi returning visitor).
- Revenue perusahaan sebelum dan sesudah pemodelan.

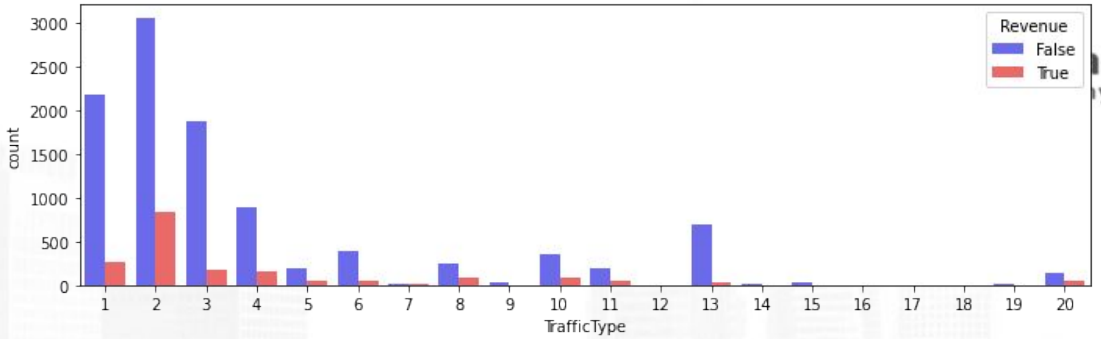
Pre-processing (1-4 slide)

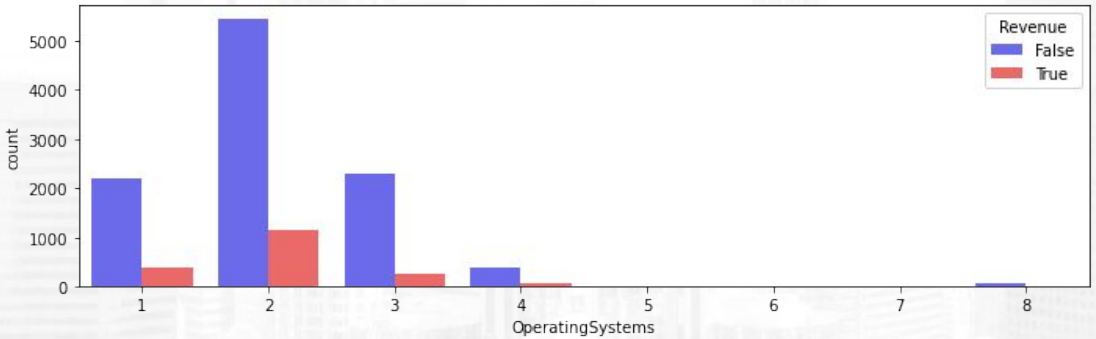
Distribusi Data

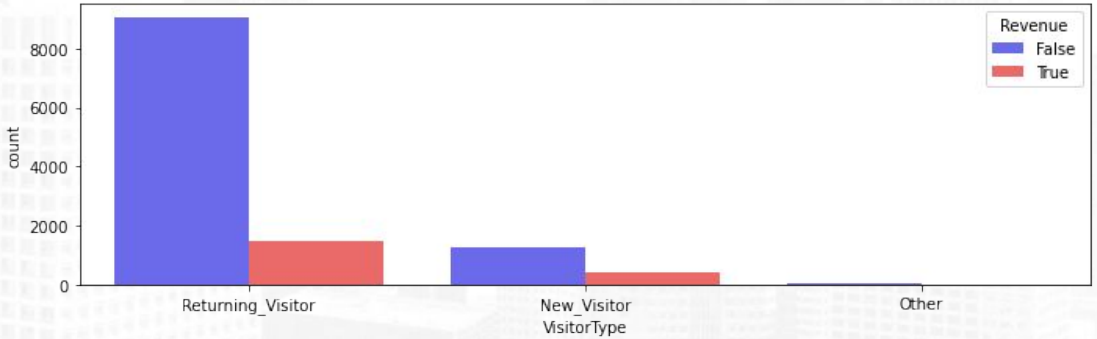


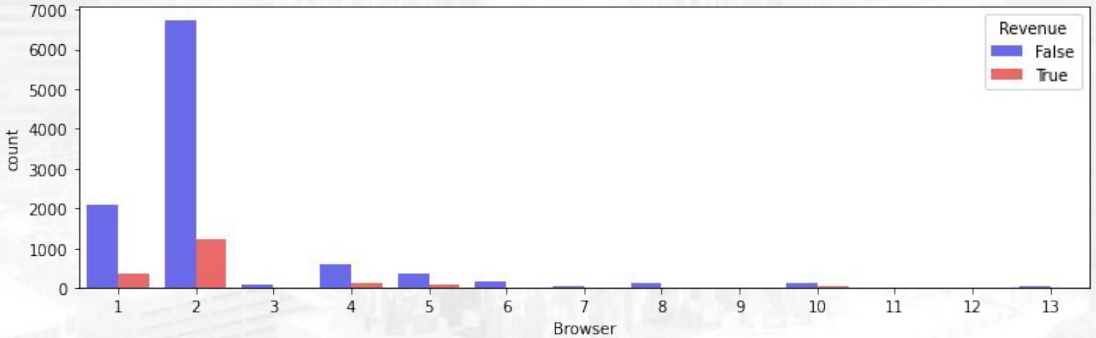
Berdasarkan hasil kde plot terlihat bahwa feature-feature yang ada positively skew

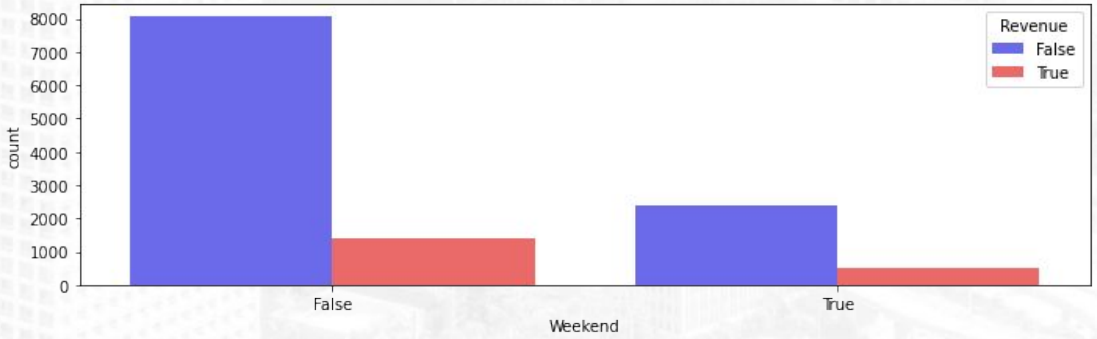

Revenue
False
True
count
Month
Feb
Mar
May
Oct
June
Jul
Aug
Nov
Sep
Dec

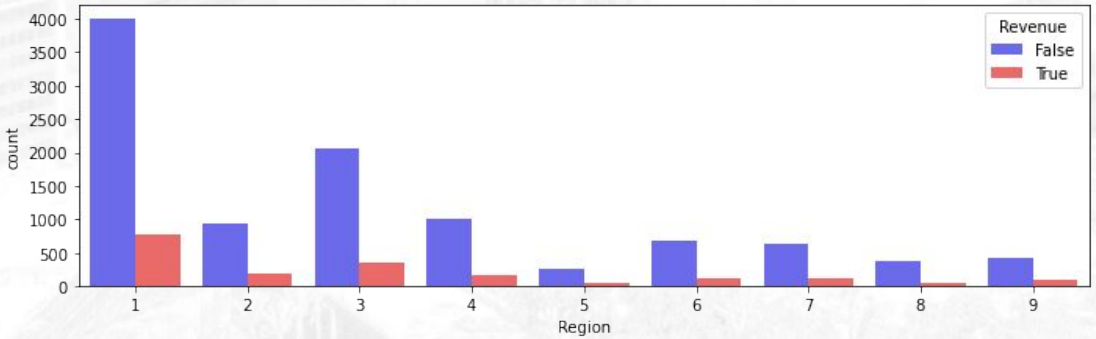

Revenue
False
True
count
TrafficType


Revenue
False
True
count
OperatingSystems


Revenue
False
True
count
Returning_Visitor
New_Visitor
Other
VisitorType

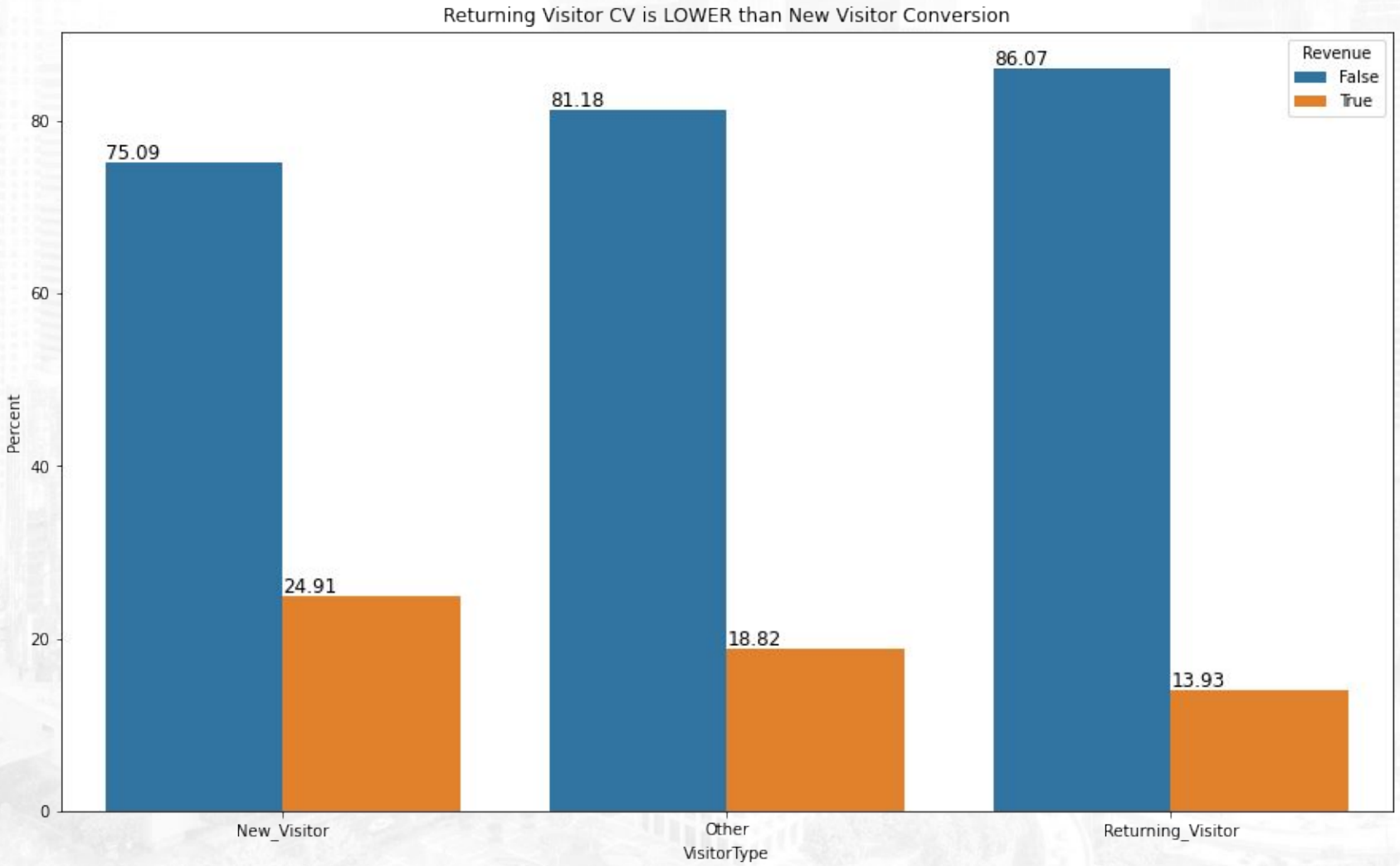

Revenue
False
True
count
Browser


Revenue
False
True
count
False
True
Weekend


Revenue
False
True
count
Region


Revenue
False
True
count
False
True
Revenue

Who is OTHER Visitor? Robot? Why is it Buying?



EXTERNAL INSIGHT

anderson-review.ucla.edu

Research Brief

Search Fatigue: Online Shoppers Grow Weary, Take a Break

A shopping session may unfold over a period of weeks, not minutes

Abundant choice is both a feature and a bug of online shopping. Being able to comparison shop with merely a click has revolutionized how we buy stuff. But when a single shopping site offers, say, hundreds of shoe choices— never mind shopping across multiple sites — choice can be exhausting.

A working paper suggests that many consumers have developed a coping mechanism: We take breaks. Not minutes, like a quick trip to the mall's food court, but days and weeks. These "search gaps" can restore our shopping mojo, and even slightly increase the likelihood of finally making a purchase.

High Level Approach: How Might We Improve This?

1. Berfokus pada segmen Returning Visitor, karena CvR nya jauh dibawah New Visitor
2. Mencari faktor yang paling berpengaruh terhadap pembelian konsumen
3. Membuat model machine learning untuk memprediksi niat beli user
4. Memberikan treatment khusus pada user yang tidak berniat membeli dengan melihat history dalam appsnya lebih lanjut
5. Treatment bisa dicluster berdasarkan probabilitas pembelian,
   a. User dengan minat tinggi beli (90%) -> Fast Checkout
   b. User dengan minat beli sedang (50%) -> Tracking Journey
   c. User dengan minat beli rendah (<50%) -> Random notifications (Promo, Products)
6. Mempercepat time to checkout customer agar tidak mengalami search fatigue sebelum pembelian

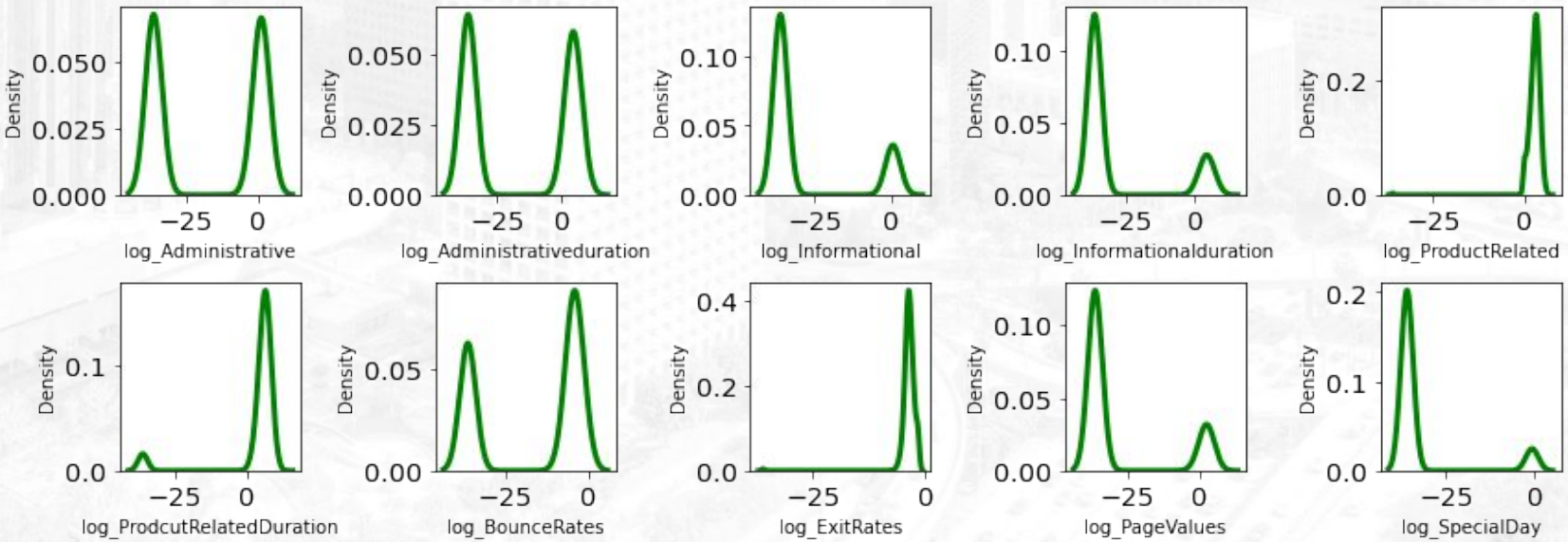
Pre-processing (1-4 slide)

Gambar sebelumnya menunjukkan distribusi data antara kolom feature dengan kolom target. Dari visualisasi tersebut ada beberapa insight yang kita bisa ambil di antaranya:

- Mei dan November bulan dengan banyak penjualan
- Tingkat Conversion rate Mei rendah dibandingkan dengan November
- Jumlah returning visitor banyak namun conversion rate rendah
- Pengunjung lebih banyak melakukan transaksi saat Weekdays
- Terdapat class imbalance pada kolom target Revenue

Data Cleansing

- Tidak terdapat data kosong
- Terdapat 125 duplicate data, namun tidak di drop karena jumlahnya lebih dari 1% jumlah data
- Drop nilai New_Visitor pada kolom VisitorType dikarenakan populasi yang dipakai adalah returning visitor, untuk 'Others' diasumsikan sebagai Returning Visitor
- Melakukan log transformation untuk feature dengan type numerik
- dilakukan handling outliers menggunakan Z-Score didapatkan jumlah baris setelah memfilter outliers adalah 9883
-



Data Transformation

Pada saat melakukan transformasi log, dilakukan penambahan epsilon dikarenakan hasil yang dihasilkan tanpa epsilon menunjukkan minus infinite.

Dengan Epsilon

```
df['log_Administrative']= np.log(df['Administrative']+np.finfo(float).eps)
df['log_Administrativeduration']= np.log(df['Administrative_Duration']+np.finfo(float).eps)
df['log_Informational']= np.log(df['Informational']+np.finfo(float).eps)
df['log_Informationalduration']= np.log(df['Informational_Duration']+np.finfo(float).eps)
df['log_ProductRelated']= np.log(df['ProductRelated']+np.finfo(float).eps)
df['log_ProdcutRelatedDuration']= np.log(df['ProductRelated_Duration']+np.finfo(float).eps)
df['log_BounceRates']= np.log(df['BounceRates']+np.finfo(float).eps)
df['log_ExitRates']= np.log(df['ExitRates']+np.finfo(float).eps)
df['log_PageValues']= np.log(df['PageValues']+np.finfo(float).eps)
df['log_SpecialDay']= np.log(df['SpecialDay']+np.finfo(float).eps)
```

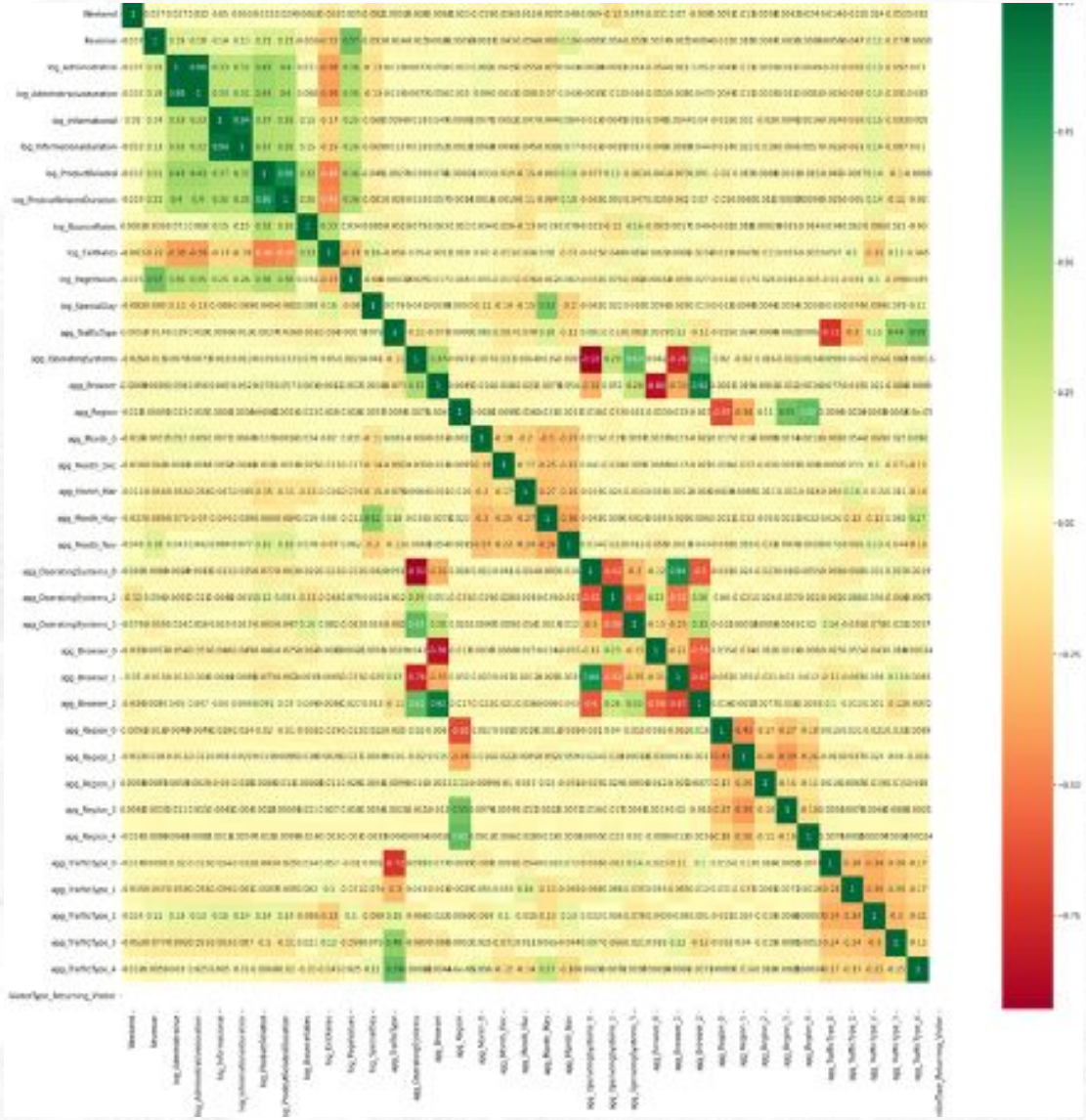
›g_Administrative	log_Administrativeduration	log_Informational	log_Informationalduration	log_ProductRelated	log_ProdcutRelatedDuratior
1.063600e+04	10636.000000	10636.000000	10636.000000	10636.000000	10636.000000
-1.724544e+01	-16.107937	-27.903292	-27.880493	2.709988	3.574355
1.862859e+01	20.242142	15.251697	16.226084	2.455661	10.709998
-3.604365e+01	-36.043653	-36.043653	-36.043653	-36.043653	-36.043653
-3.604365e+01	-36.043653	-36.043653	-36.043653	1.945910	5.241747
2.220446e-16	-36.043653	-36.043653	-36.043653	2.944439	6.478561
1.386294e+00	4.477337	-36.043653	-36.043653	3.713572	7.374288
3.295837e+00	8.131163	3.178054	7.843604	6.558198	11.066225

Tanpa Epsilon

df[‘log_Administrative’]=np.log(df[‘Administrative’])

log_Administrative1
1.063600e+04
-inf
NaN
-inf
NaN
0.000000e+00
1.386294e+00
3.295837e+00

Feature Selection



Pada tahap ini feature yang dipilih untuk tahap selanjutnya ialah yang memilki nilai korelasi terhadap target >0,05 dengan tanpa feature redundant yaitu

1. log_Administrative
2. log_Informational
3. log_ProductRelated
4. log_ExitRates
5. log_PageValues
6. agg_Month_Nov
7. agg_OperatingSystems_2
8. agg_TrafficType_2

Feature Encoding

Dilakukan feature encoding, karena ada data yang berbentuk kategorikal dan ada beberapa data yang memiliki kardinalitas tinggi. untuk menangani data dengan kardinalitas yang tinggi digunakan fungsi Agregasi sederhana, idenya dengan membiarkan instance milik nilai dengan frekuensi tinggi apa adanya dan ganti instance lain dengan kategori baru yang akan kita sebut other (pada dataset ini disebut 0). langkahnya:

1. Pilih ambang batas (pada data kami menggunakan 75%)
2. Urutkan nilai unik di kolom berdasarkan frekuensinya dalam urutan menurun
3. Terus tambahkan frekuensi nilai unik yang diurutkan (turun) ini hingga ambang tercapai.
4. Hasil Kategori unik pada proses 2 & 3 yang akan disimpan, dan contoh dari semua kategori lainnya akan diganti dengan 0

OHE untuk feature : TrafficType, Month, OperatingSystem, Browser, Region
Label Encoding untuk feature Weekend dan Target (Revenue)

Split Data

```
# Split Feature and Label
X = df[['log_Administrative','log_Informational','log_ProductRelated','log_ExitRates',
        'log_PageValues','agg_Month_Nov','agg_OperatingSystems_2','agg_TrafficType_2']]
y = df['Revenue'] # target / label

#Splitting the data into Train and Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)
```

Kami melakukan data split dimana variabel X memiliki 8 feature dan y memiliki satu feature. Perbandingan data train dan data test sebesar 70: 30.

Handle Class Imbalance

Setelah dilakukan label encoding terhadap feature target yaitu Revenue, maka kita baru bisa melakukan class imbalance terhadap feature tersebut. Hal ini dilakukan untuk mengoptimalisasi akurasi machine learning lebih pintar dan menyeimbangkan data minoritas.

```
df['Revenue_1'] = df['Revenue']
df['Revenue_1'].value_counts()
```

```
0    8422
1    1461
Name: Revenue_1, dtype: int64
```

```
X = df[[col for col in df.columns if (str(df[col].dtype) != 'object') and col not in ['Revenue','Revenue_1']]]
y = df['Revenue_1'].values
print(X.shape)
print(y.shape)
```

```
(9883, 37)
(9883,)
```

```
from imblearn.over_sampling import SMOTE
smote = SMOTE(random_state = 42)

X_over_SMOTE_train, y_over_SMOTE_train = smote.fit_resample(X_train, y_train)
```

```
print(pd.Series(y_over_SMOTE_train).value_counts())
```

```
0    5899
1    5899
Name: Revenue, dtype: int64
```

Pada tahap ini tidak dilakukan undersampling/oversampling karena pada hasil feature encoding perbedaan nilai setiap kolom tidak terlalu signifikan dikarenakan sudah diclustering

STAGE2 - Modelling Experiments (1-3 slide)

- Kami telah melakukan eksperimen dengan 5 algoritma pemodelan diantaranya Logistic Regression, Decision Tree, Random Forest, Adaboost dan XGboost dengan nilai recall tertinggi diperoleh oleh Logistic Regression dengan recall 80%, dan kami memutuskan untuk menggunakan pemodelan logistic regression juga dikarenakan score test dengan score train best fit
- Dengan menggunakan feature yang sudah diselect sebelumnya, recall pada Logistic Regression meningkat menjadi 82%

Hasil Modelling Experiments sebelum tuning hyperparameter

Model	Accuracy (%)	Precision (%)	Recall (%)	AUC(%)	F1(%)
Logistic Regression	86	51	80	83	62
Decision Tree	84	47	62	74	53
Random Forest	86	53	70	79	60
Adaboost	87	53	79	83	64
XGB	87	56	63	77	59

Tuning Hyperparameter

Tuning Hyperparameters

```
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV

# List Hyperparameters yang akan diuji
solver = ['newton-cg', 'lbfgs', 'liblinear']
penalty = ['l2','l1', 'elasticnet', 'none']
C = [100, 10, 1.0, 0.1, 0.01, 0.001, 0.0001]
hyperparameters = dict(penalty=penalty, C=C, solver=solver )

# Inisiasi model
logres = LogisticRegression(random_state=42) # Init Logres dengan Gridsearch, cross validation = 5
lr_tuned = RandomizedSearchCV(logres, hyperparameters, cv=5, random_state=42, scoring='recall')

# Fitting Model & Evaluation
lr_tuned.fit(X_over_SMOTE, y_over_SMOTE)
eval_classification(lr_tuned)
```

Hasil Modelling Experiments sesudah tuning hyperparameter

Model	Accuracy (%)	Precision (%)	Recall (%)	AUC(%)	F1(%)
Logistic Regression	86	52	80	84	63
Decision Tree	84	46	59	73	52
Random Forest	87	55	72	89	62
Adaboost	87	54	79	84	64
XGB	86	53	65	77	58

Menggunakan Tuning Hyperparameter

- C = [100, 10, 1.0, 0.1, 0.01, 0.001, 0.0001]
- penalty = ['l2','l1', 'elasticnet', 'none']
- solver = ['newton-cg', 'lbfgs', 'liblinear']

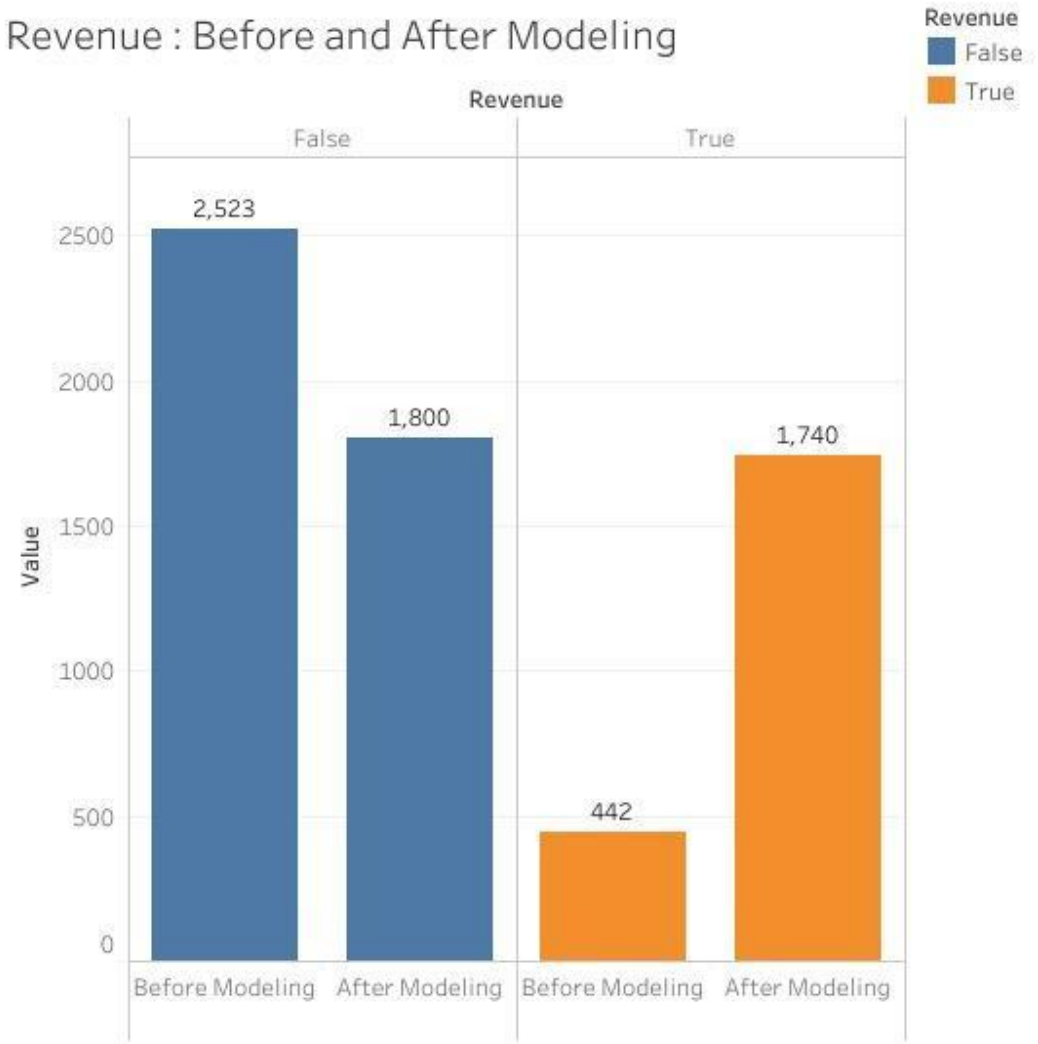
Dengan Best Params :Random State = 42, C = 100, Solver = newton-cg

Hasil modelling ulang dengan feature selection menggunakan pemodelan Logistic Regression

Model	Accuracy	Precision	Recall	AUC	F1
Logistic Regression	84	85	82	84	84

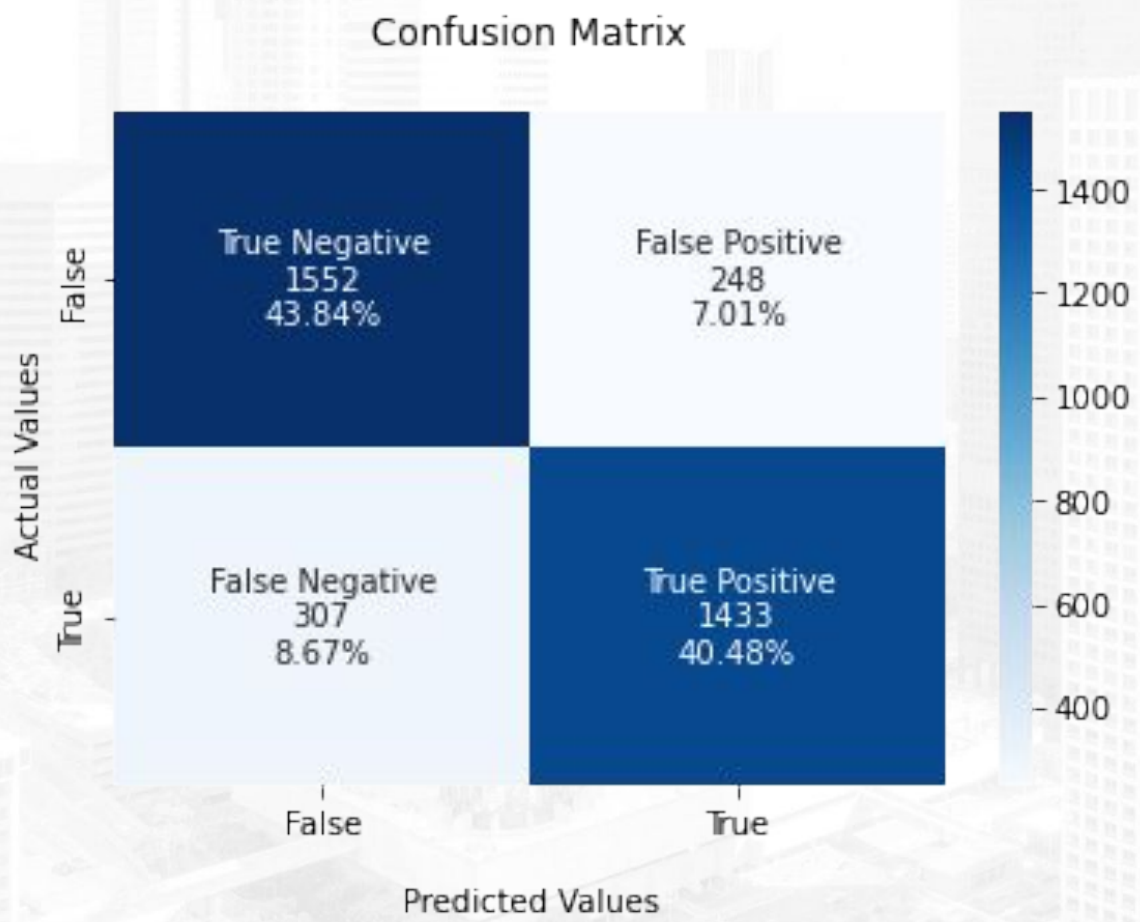
Kami melakukan modelling ulang dengan 6 feature dari score tertinggi feature importance pemodelan Logistic Regression yaitu feature Log Page Values, Agg Month Nov, Log Product Related, Agg Traffic Type 2, Log Informational, dan Log Administrative

RESULT MODELING



Dengan memberikan treatment yang tepat terhadap Returning Visitor yang diprediksi mempunyai intensi membeli, jumlah visitor yang convert mengalami kenaikan sebesar 1298.

Confusion Matrix



Dengan kenaikan jumlah visitor yang convert sebesar 75% kemudian dilakukan perhitungan terhadap profit awal sehingga didapatkan kenaikan profit sebesar 26%

Asumsi **profit** e-Commerce :
Rp. 20.000.000/hari

	Profit/hari	Revenue
Before	Rp. 20.000.000	15%
After	Rp. 35.000.000	26%

High Level Approach: How Might We Improve This?

1. Berfokus pada segmen Returning Visitor, **karena CRnya jauh dibawah** New Visitor
2. Mencari **faktor yang paling berpengaruh** terhadap pembelian konsumen
3. Membuat model machine learning **untuk memprediksi niat beli user**
4. Memberikan **treatment khusus pada user yang tidak berniat membeli** dengan melihat journey dalam appsnya lebih lanjut (rekomendasi kedepan)
5. Treatment bisa **dicluster berdasarkan probabilitas pembelian**,
    a. User dengan minat tinggi beli (90%) -> Fast Checkout
    b. User dengan minat beli sedang (50%) -> Tracking Journey
    c. User dengan minat beli rendah (<50%) -> Random notifications (Promo, Products)
6. Mempercepat time to checkout customer agar tidak mengalami search fatigue sebelum pembelian

Added Insight

Dari feature importance yang dihasilkan saat modelling, maka disarankan perusahaan untuk:

1. Mempelajari page dengan Page Value yang tinggi.
2. Traceback metode marketing pada bulan November dan May, apa yang membuat serbuan pengunjung dan konversi.
3. Mengetahui traffic pembeli yang kurang produktif, yaitu selain dari Operating System 2 dan Traffic Type 2.
4. Mengecek payment method pada page administrative, serta information targeting yang tepat pada information page (sesuai minat customer).

Execution Guidelines

- Perusahaan bisa melakukan A/B testing kepada segmen customer yang “True Negative” namun melakukan kunjungan ulang pada website
- A/B testing merupakan outcomes dari formulasi fitur-fitur penting dalam melakukan pembelian, misal :

  A. Memberikan stimulus page, serupa dengan page dengan pageValue yang tinggi

  B. Memberikan kampanye serupa yang dilakukan pada bulan November/May

EXTERNAL INSIGHT

- Berdasarkan anderson-review.ucla.edu banyak nya pilihan pencarian pada suatu produk dapat menyebabkan *fatigue* pada online shopper yang mengunjungi website.
- Sebuah penelitian di Belanda terhadap lebih dari 4600 online shopper menemukan bahwa lebih dari 40% nya memilih untuk 'istirahat' sebelum kembali berbelanja.
- Para peneliti membangun model yang memberi online shopper pilihan untuk berbelanja sekarang atau nanti.
- Para peneliti memperkirakan jika *fatigue* online shopper dapat berkurang setengah nya, pembelian bisa meningkat sekitar 1%

EXTERNAL INSIGHT

- Berdasarkan karakteristik hasil pemodelan, perusahaan perlu melakukan **treatment** lebih terhadap visitor yang yang **diprediksi tidak membeli** seperti promo gratis ongkir/potongan harga sebelum visitor mengalami *fatigue* dan memutuskan untuk istirahat.
- Perusahaan dapat mengimplementasikan model ini dalam **memprediksi intensi** visitor membeli atau tidak.
- Perusahaan dapat lebih **meningkatkan** halaman Product Related, Informational dan Administrative supaya meningkatkan *user experience* dalam berbelanja sehingga mencegah *fatigue* visitor.

Expected Return

1. Mid-Scale E-Commerce like Tokopaedi **made 6 T bruto return/year**
2. There are **75% users that counts as very unlikely to buy**
3. If we can make users that unlikely to buy, “considering”, and 10% of them buy, **then our profit will rose 50% to be 9 T**

Pembagian Tugas (1 slide)

Penjelasan kontribusi dari masing-masing anggota team (siapa mengerjakan apa saja). Tuliskan secara detail, termasuk pembagian dalam penulisan laporan, pembuatan slide, dan presentasi.

- Notulen : Inggriani, Desy
- Coding : Donny, Maya, Arif, Desy
- PPT presentasi : Inggriani, Danang
- Laporan : Desy, Maya

Pada praktiknya setiap anggota berkontribusi di setiap tahap pengerjaan baik itu coding, penyusunan materi presentasi dan laporan. Namun kami membagi PIC disetiap tahap supaya lebih mudah dalam koordinasi dan follow up progress