Predict Clicked Ads Customer Classification by using Machine Learning

Created by:
Danu Satria Wiratama
danusatria06@gmail.com
https://www.linkedin.com/in/danusatria/

"I am a computer science graduate from Satya Wacana Christian University with a concentration in data science. I completed an online bootcamp at Rakamin Academy with a topic of data science, which further enhance my skills in data analysis and machine learning. I have mastered several skills such as SQL, python, data visualization, and additional proficiency in machine learning for further modeling and analysis.. "

Supported by:
Rakamin Academy
Career Acceleration School
www.rakamin.com

Overview

“Sebuah perusahaan di Indonesia ingin mengetahui efektifitas sebuah iklan yang mereka tayangkan, hal ini penting bagi perusahaan agar dapat mengetahui seberapa besar ketercapainnya iklan yang dipasarkan sehingga dapat menarik customers untuk melihat iklan.

Dengan mengolah data historical advertisement serta menemukan insight serta pola yang terjadi, maka dapat membantu perusahaan dalam menentukan target marketing, fokus case ini adalah membuat model machine learning classification yang berfungsi menentukan target customers yang tepat ”

Exploratory Data Analysis (EDA)

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Unnamed: 0                1000 non-null   int64  
 1   Daily Time Spent on Site  987 non-null    float64
 2   Age                       1000 non-null   int64  
 3   Area Income               987 non-null    float64
 4   Daily Internet Usage      989 non-null    float64
 5   Male                      997 non-null    object 
 6   Timestamp                 1000 non-null   object 
 7   Clicked on Ad             1000 non-null   object 
 8   city                      1000 non-null   object 
 9   province                  1000 non-null   object 
 10  category                  1000 non-null   object 
dtypes: float64(3), int64(2), object(6)
memory usage: 86.1+ KB
```

```
df.shape
```

```
(1000, 11)
```

Untuk selengkapnya, dapat melihat jupyter notebook disini

- Dataset ini terdiri dari 11 kolom dengan 1000 baris.
- Unnamed : Kolom pengenal dengan 1000 nilai integer non-null.
- Daily Time Spent on Site : Berisi 987 nilai float64 non-null, yang mewakili rata-rata waktu harian yang dihabiskan di situs web.
- Age : Berisi 1000 nilai integer non-null, yang berisi usia pengguna.
- Area Income : Berisi 987 nilai float64 non-null, yang mewakili tingkat pendapatan pengguna di berbagai area.
- Daily Internet Usage : Berisi 989 nilai float64 non-null, yang mewakili rata-rata penggunaan internet harian oleh pengguna.
- Male : Berisi 997 nilai objek non-null, yang menunjukkan jenis kelamin pengguna.
- Timestamp : Berisi 1000 nilai objek non-null, yang menunjukan stempel tanggal dan waktu.
- Clicked on Ad : Berisi 1000 nilai objek non-null, yang menunjukkan apakah pengguna mengklik iklan ("Ya" atau "Tidak").
- City : Berisi 1000 nilai objek non-null, yang menunjukkan kota tempat pengguna berada.
- Province : Berisi 1000 nilai objek non-null, yang menunjukkan provinsi atau kawasan pengguna.
- Category : Berisi 1000 nilai objek non-null, yang mewakili variabel atau kategori kategoris.

Customer Type and Behaviour Analysis on Advertisement

Rakamin Academy

- Statistical Analysis pada kolom numerik

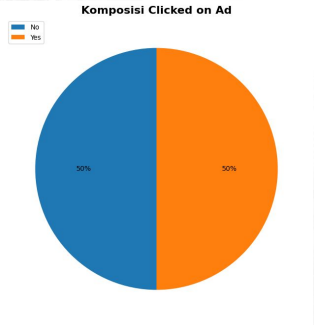
	count	mean	std	min	25%	50%	75%	max
Daily Time Spent on Site	987.0	6.492952e+01	1.584470e+01	32.60	5.127000e+01	6.811000e+01	7.846000e+01	9.143000e+01
Age	1000.0	3.600900e+01	8.785562e+00	19.00	2.900000e+01	3.500000e+01	4.200000e+01	6.100000e+01
Area Income	987.0	3.848647e+08	9.407999e+07	97975500.00	3.286330e+08	3.990683e+08	4.583554e+08	5.563936e+08
Daily Internet Usage	989.0	1.798636e+02	4.387014e+01	104.78	1.387100e+02	1.826500e+02	2.187900e+02	2.670100e+02

- Statistical Analysis pada kolom kategori

	count	unique	top	freq
Male	997	2	Perempuan	518
city	1000	30	Surabaya	64
province	1000	16	Daerah Khusus Ibukota Jakarta	253
category	1000	10	Otomotif	112

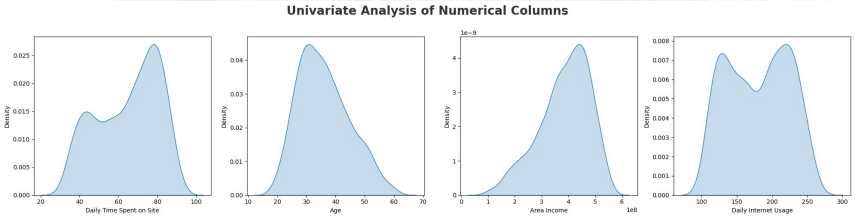
Univariate Analysis

- Komposisi target (Clicked on Ad)



Persebaran data pada target (Clicked on Ad) sama ratanya, sehingga tidak diperlukan pemrosesan untuk mengurangi / menambah data.

- Univariate Analysis pada kolom numerik

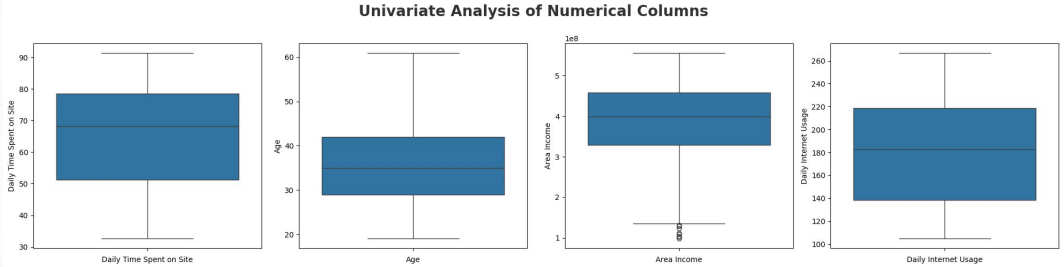



```
Skewness Daily Time Spent on Site : -0.370
Skewness Age : 0.479
Skewness Area Income : -0.644
Skewness Daily Internet Usage : -0.031
```

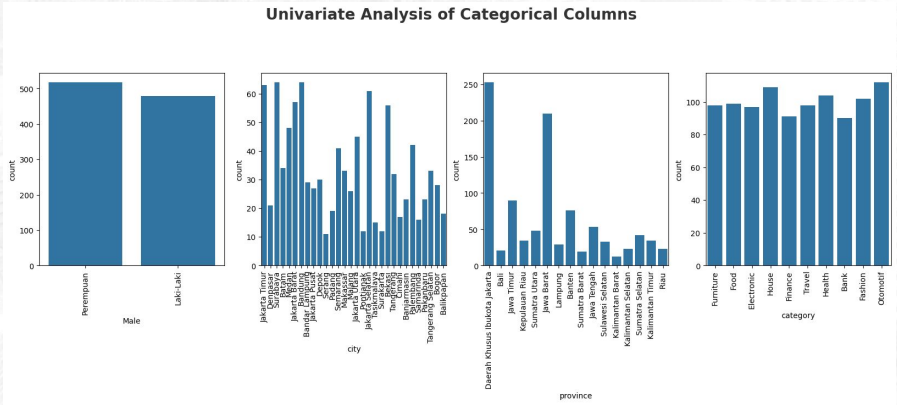
* Kolom Daily Time Spent on Site, Age, Daily Internet Usage memiliki distribusi yang hampir normal.
* Kolom Area Income memiliki negative skewed yang mengindikasi adanya outlier.

Univariate Analysis

- Univariate Analysis pada kolom numerik

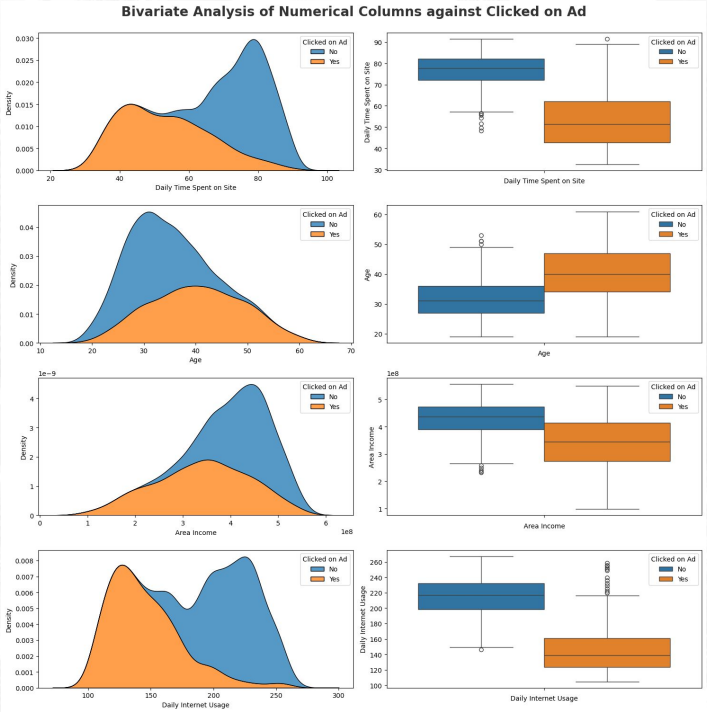


- Univariate Analysis pada kolom kategori

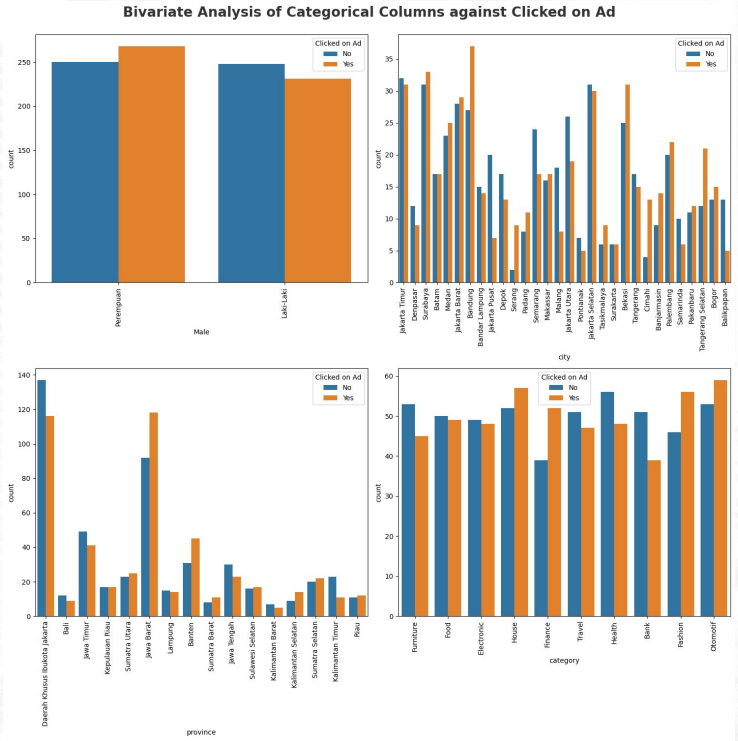


Bivariate Analysis

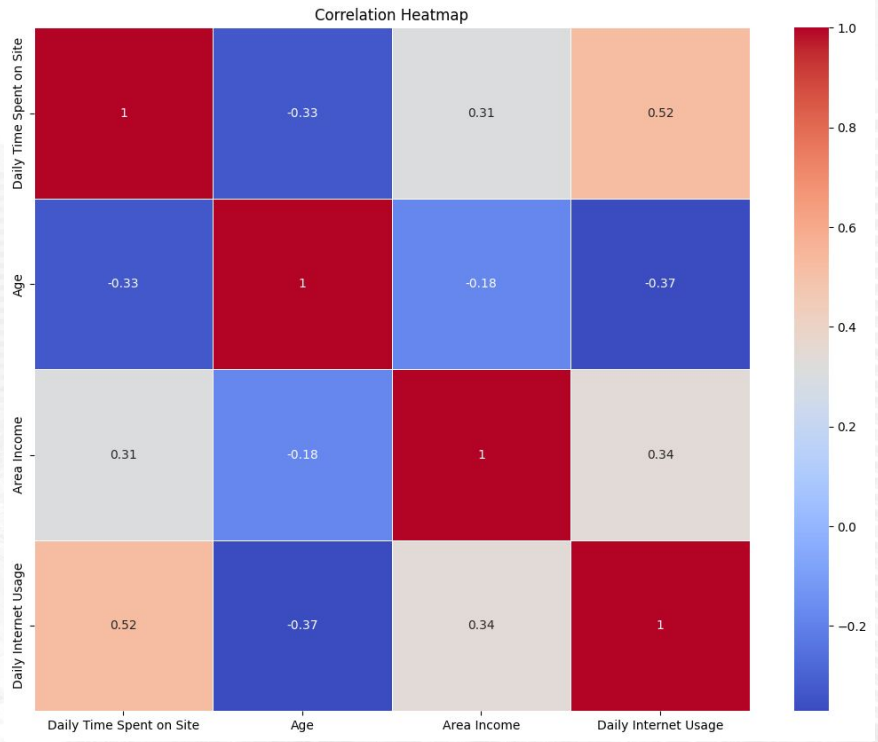
Bivariate Analysis pada kolom numerik berdasarkan ‘Cliked on Ad’



Bivariate Analysis pada kolom kategori berdasarkan ‘Cliked on Ad’



Multivariate Analysis



- Daily Time Spent on Site dan Daily Internet Usage memiliki korelasi positif yang cukup kuat. Artinya, semakin banyak waktu yang dihabiskan seseorang di situs web, semakin besar kemungkinan mereka memiliki penggunaan internet harian yang tinggi.

- Daily Time Spent on Site dan Age memiliki korelasi negatif yang cukup kuat. Artinya berarti semakin tinggi usia seseorang, semakin sedikit waktu yang mereka habiskan di situs web.

Conclusion

- Customer yang klik iklan memiliki rata-rata waktu mengunjungi layar antara 40-50 menit.
- Umur customer yang klik iklan di antara range 35-45 tahun.
- Customer dengan pendapatan area diantara 2.5-4 cenderung untuk klik iklan.
- Customer dengan penggunaan 120-160 cenderung untuk klik iklan.
- Perempuan lebih banyak klik iklan dibandingkan laki-laki.
- Kota Bandung memiliki tingkat klik iklan yang paling tinggi, sedangkan yang paling rendah di Kota Serang.
- Provinsi Jawa Barat memiliki tingkat klik iklan yang paling tinggi, sedangkan yang paling rendah di Provinsi Kalimantan Barat.
- Kategori yang paling diminati adalah otomotif, sedangkan yang paling tidak diminati adalah finance.

Data Cleaning & Preprocessing

Terdapat 13 data null, data tersebut sangat kecil dibandingkan keseluruhan data (1,3%). Sehingga dilakukan penghapusan.

```
df_clean.isnull().sum().sort_values(ascending=False)
```

	0
Daily Time Spent on Site	13
Area Income	13
Daily Internet Usage	11
Male	3
Unnamed: 0	0
Age	0
Timestamp	0
Clicked on Ad	0
city	0
province	0
category	0

	0
Unnamed: 0	0
Daily Time Spent on Site	0
Age	0
Area Income	0
Daily Internet Usage	0
Male	0
Timestamp	0
Clicked on Ad	0
city	0
province	0
category	0

```
df_clean.duplicated().sum()
```

```
0
```

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Cleaning & Preprocessing

Rakamin Academy

- Extract Datetime data

```
# Mengubah 'Timestamp' kolom ke datetime format
df_clean['Timestamp'] = pd.to_datetime(df_clean['Timestamp'])

# Ekstrak tahun, bulan, minggu, dan hari di kolom terpisah
df_clean['Year'] = df_clean['Timestamp'].dt.year
df_clean['Month'] = df_clean['Timestamp'].dt.month
df_clean['Week'] = df_clean['Timestamp'].dt.isocalendar().week
df_clean['Day'] = df_clean['Timestamp'].dt.day

df_clean.drop(columns=['Timestamp'], axis=1, inplace=True)
```

- Melakukan feature encoding pada kolom 'Gender', 'Clicked on Ad', 'city', 'province', 'category'

```
onehot = ['Gender', 'Clicked on Ad', 'city', 'province', 'category']

# One-hot encoding
for cats in onehot:
    onehots = pd.get_dummies(df_clean[cats], prefix=cats)
    df_clean = df_clean.join(onehots)

df_clean.drop(columns=onehot, axis=1, inplace=True)
```

Data Cleaning & Preprocessing

- Melakukan pembagian data fitur dan target

```
# Split data fitur dan target

X = df_clean.drop(['Clicked on Ad_Yes'], axis=1)

y = df_clean['Clicked on Ad_Yes']
```

Data Modeling

- Membagi data training dan testing

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

print("Data Split Details:")
print(f"Total Samples: {len(X)}")
print(f"Training Samples: {len(X_train)} ({len(X_train)/len(X)*100:.2f}%)")
print(f"Testing Samples: {len(X_test)} ({len(X_test)/len(X)*100:.2f}%)")
```

```
Data Split Details:
Total Samples: 963
Training Samples: 674 (69.99%)
Testing Samples: 289 (30.01%)
```

Dari 963 data, dibagi menjadi 674 sebagai data training dan 289 sebagai data testing

- Melakukan eksperimen dengan berbagai model machine learning (dengan normalisasi dan tanpa normalisasi)

Hasil Evaluasi Model Tanpa Normalisasi:

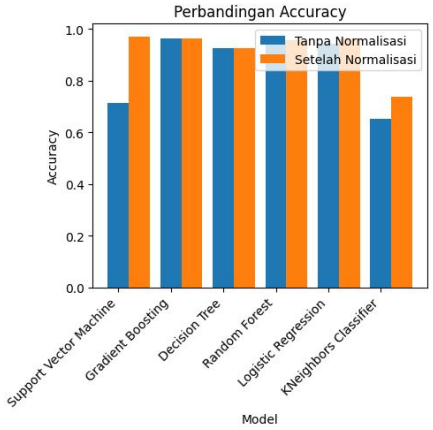
	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Support Vector Machine	0.712803	0.787037	0.586207	0.671937	0.713242
1	Gradient Boosting	0.965398	0.953020	0.979310	0.965986	0.965350
2	Decision Tree	0.927336	0.907895	0.951724	0.929293	0.927251
3	Random Forest	0.958478	0.940397	0.979310	0.959459	0.958405
4	Logistic Regression	0.944637	0.938776	0.951724	0.945205	0.944612
5	KNeighbors Classifier	0.653979	0.671756	0.606897	0.637681	0.654143

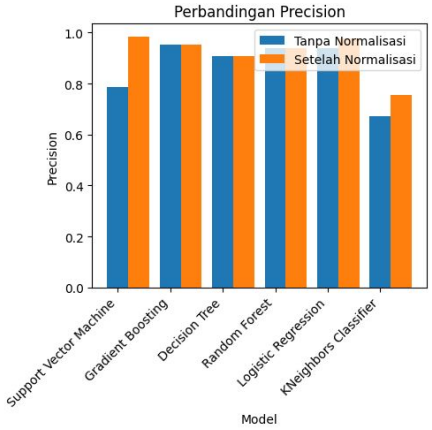
Hasil Evaluasi Model Setelah Normalisasi:

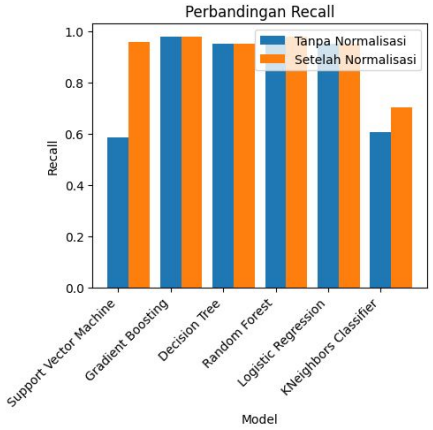
	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Support Vector Machine	0.972318	0.985816	0.958621	0.972028	0.972366
1	Gradient Boosting	0.965398	0.953020	0.979310	0.965986	0.965350
2	Decision Tree	0.927336	0.907895	0.951724	0.929293	0.927251
3	Random Forest	0.958478	0.940397	0.979310	0.959459	0.958405
4	Logistic Regression	0.965398	0.978723	0.951724	0.965035	0.965445
5	KNeighbors Classifier	0.737024	0.755556	0.703448	0.728571	0.737141

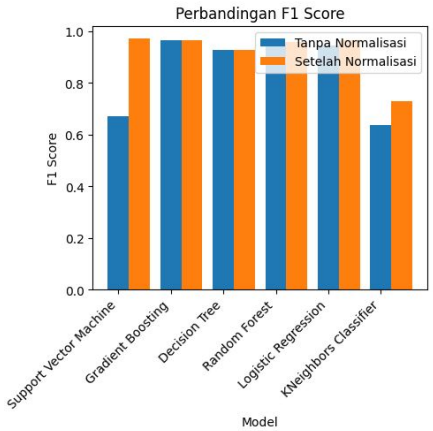
Untuk selengkapnya, dapat melihat jupyter notebook disini

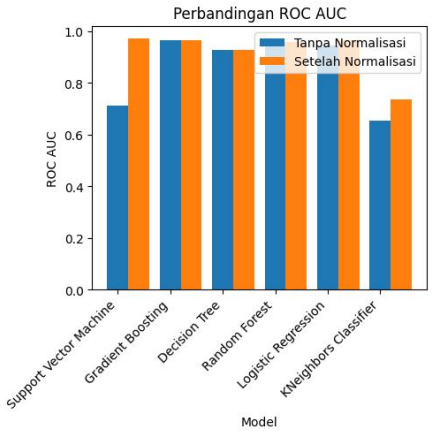
Data Modeling





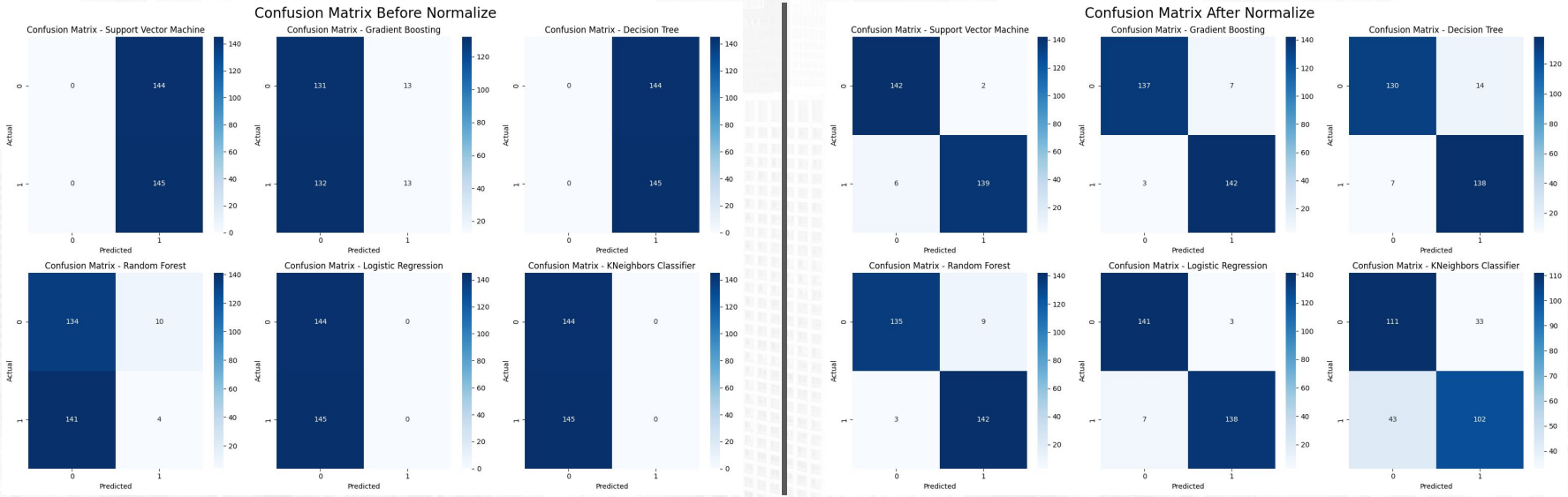






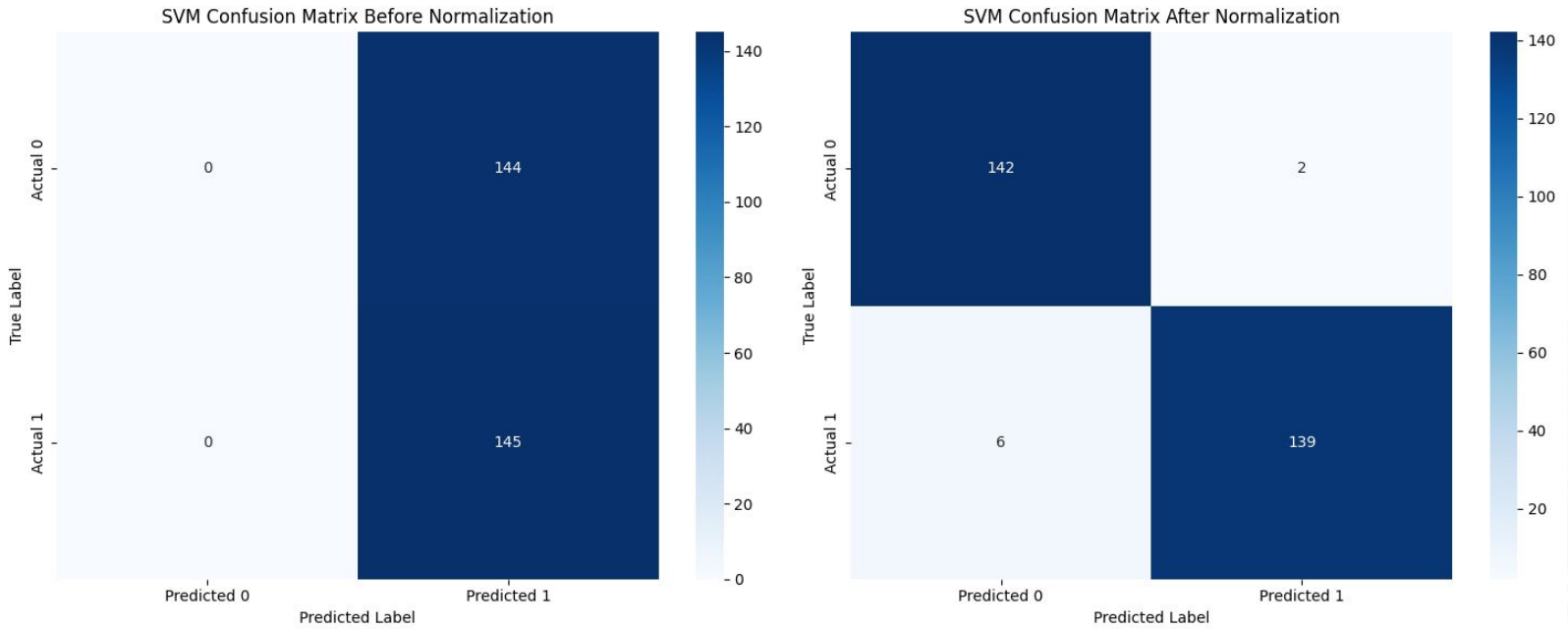
Berdasarkan grafik tersebut, terlihat bahwa proses modeling yang dilakukan normalisasi memiliki hasil yang lebih baik dibandingkan tanpa melalui proses normalisasi

Data Modeling



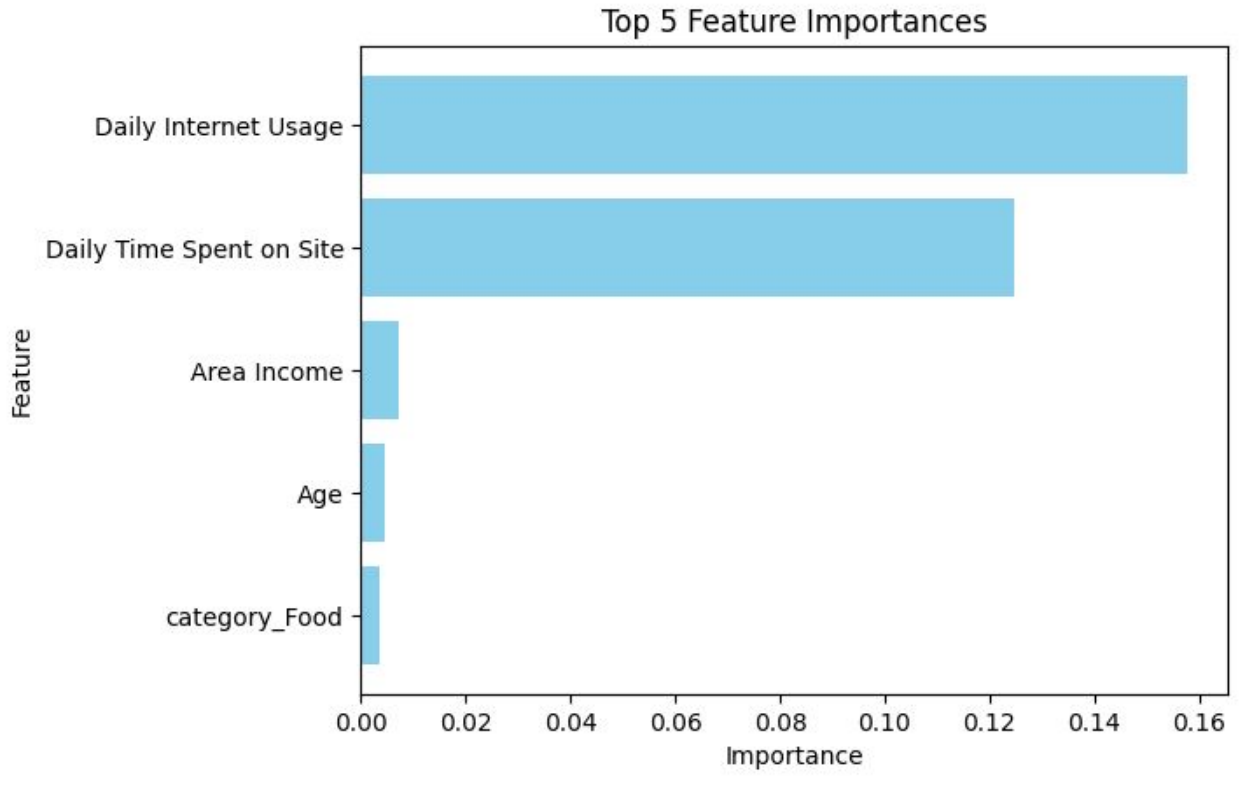
Confusion Matrix Before Normalization vs Confusion Matrix After Normalization

Data Modeling



Model SVM adalah model dengan performa yang paling bagus

Data Modeling



Data Modeling

- Membagi data training dan testing

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

print("Data Split Details:")
print(f"Total Samples: {len(X)}")
print(f"Training Samples: {len(X_train)} ({len(X_train)/len(X)*100:.2f}%)")
print(f"Testing Samples: {len(X_test)} ({len(X_test)/len(X)*100:.2f}%)")
```

```
Data Split Details:
Total Samples: 963
Training Samples: 674 (69.99%)
Testing Samples: 289 (30.01%)
```

Dari 963 data, dibagi menjadi 674 sebagai data training dan 289 sebagai data testing

- Melakukan eksperimen dengan berbagai model machine learning (dengan normalisasi dan tanpa normalisasi)

Hasil Evaluasi Model Tanpa Normalisasi:

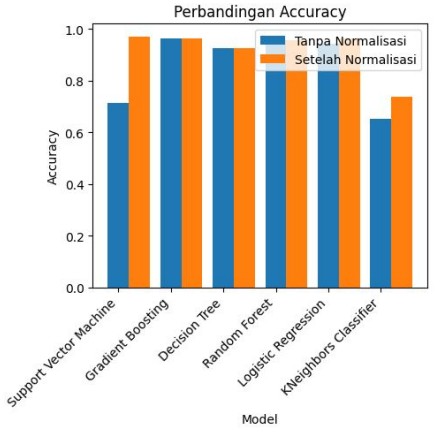
	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Support Vector Machine	0.712803	0.787037	0.586207	0.671937	0.713242
1	Gradient Boosting	0.965398	0.953020	0.979310	0.965986	0.965350
2	Decision Tree	0.927336	0.907895	0.951724	0.929293	0.927251
3	Random Forest	0.958478	0.940397	0.979310	0.959459	0.958405
4	Logistic Regression	0.944637	0.938776	0.951724	0.945205	0.944612
5	KNeighbors Classifier	0.653979	0.671756	0.606897	0.637681	0.654143

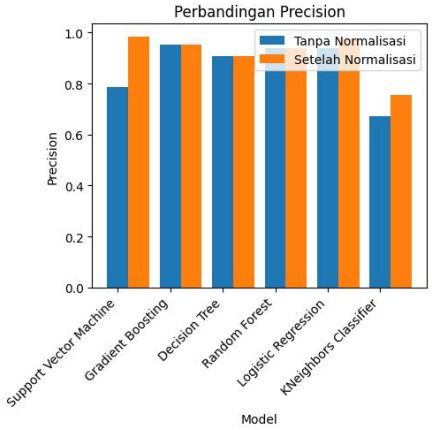
Hasil Evaluasi Model Setelah Normalisasi:

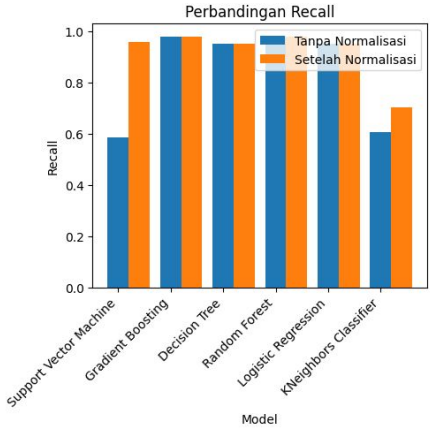
	Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
0	Support Vector Machine	0.972318	0.985816	0.958621	0.972028	0.972366
1	Gradient Boosting	0.965398	0.953020	0.979310	0.965986	0.965350
2	Decision Tree	0.927336	0.907895	0.951724	0.929293	0.927251
3	Random Forest	0.958478	0.940397	0.979310	0.959459	0.958405
4	Logistic Regression	0.965398	0.978723	0.951724	0.965035	0.965445
5	KNeighbors Classifier	0.737024	0.755556	0.703448	0.728571	0.737141

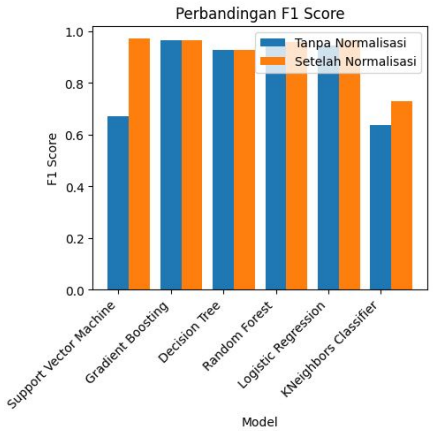
Untuk selengkapnya, dapat melihat jupyter notebook disini

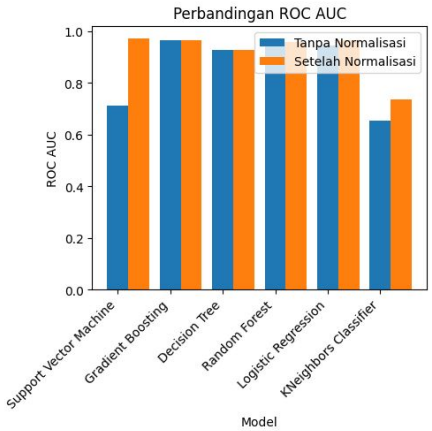
Data Modeling





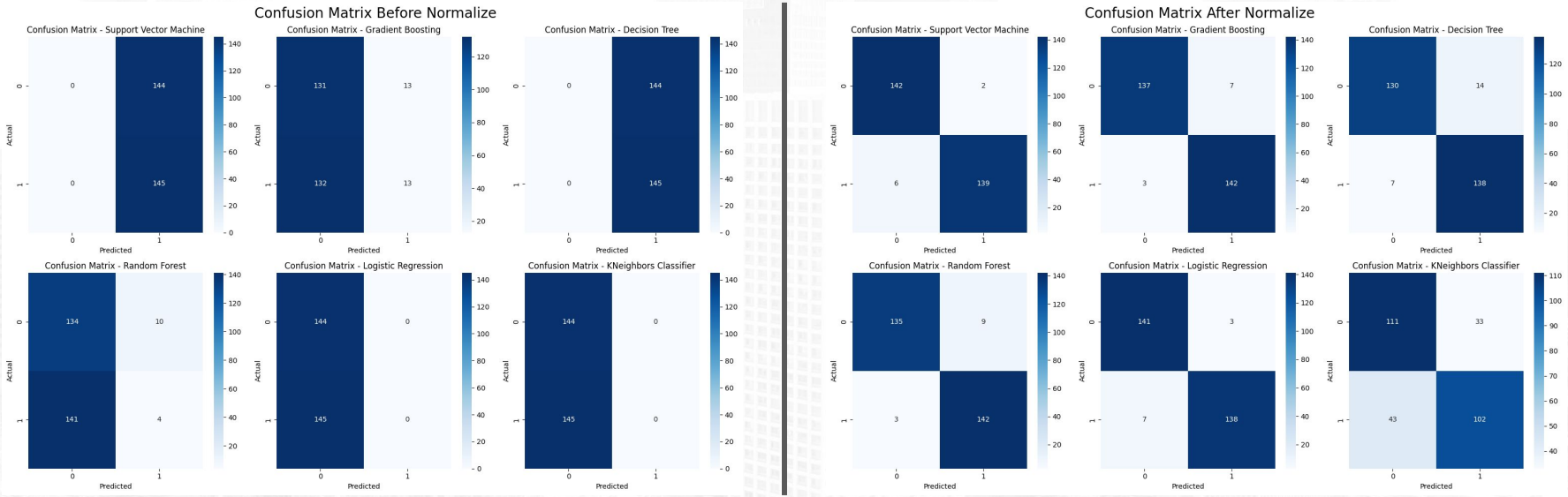






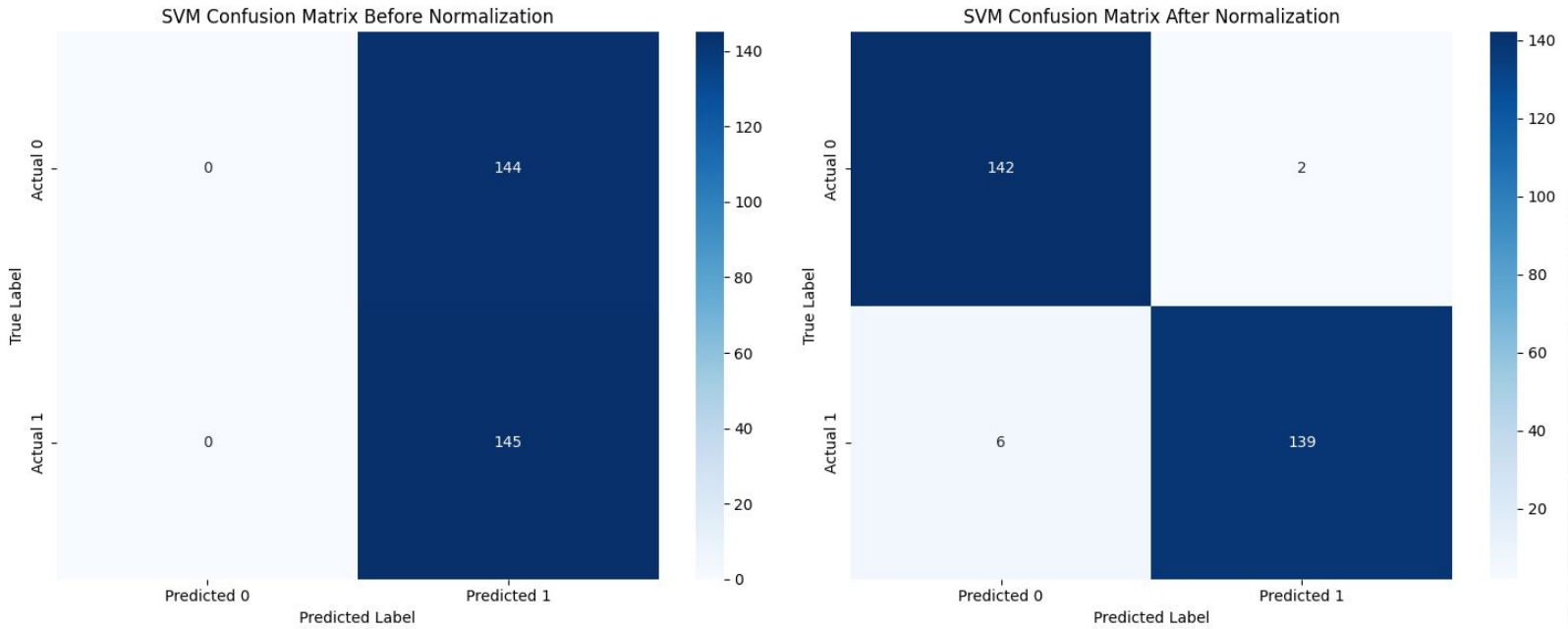
Berdasarkan grafik tersebut, terlihat bahwa proses modeling yang dilakukan normalisasi memiliki hasil yang lebih baik dibandingkan tanpa melalui proses normalisasi

Data Modeling



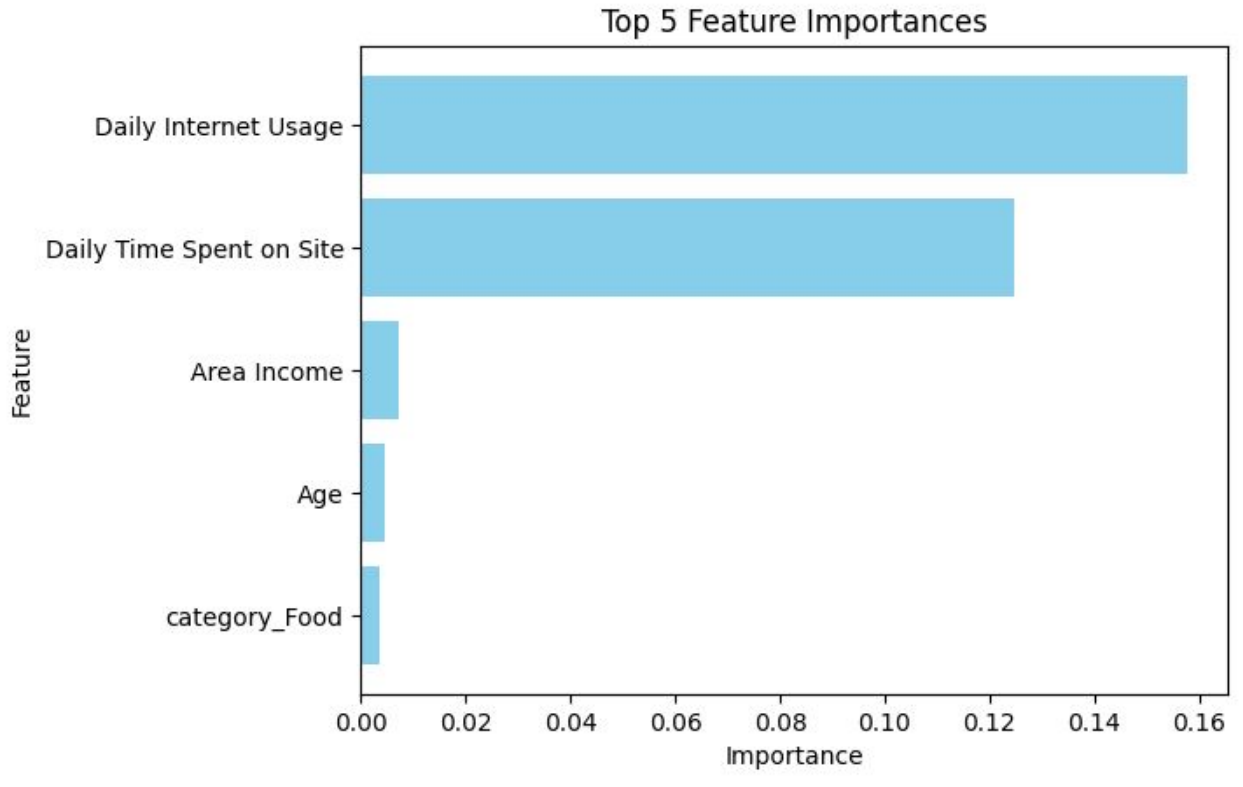
Confusion Matrix Before Normalization vs Confusion Matrix After Normalization

Data Modeling



Model SVM adalah model dengan performa yang paling bagus

Data Modeling



Business Recommendation

Business Recommendation & Simulation

- Optimalkan Pengalaman Pengguna : Tingkatkan kualitas konten dan desain situs web untuk meningkatkan waktu yang dihabiskan pengguna di situs. Gunakan data pengguna untuk memberikan rekomendasi produk atau layanan yang relevan.
- Segmentasi Pasar Berdasarkan Umur : Kelompok usia lanjut cenderung lebih responsif terhadap iklan. Dengan demikian, kita dapat mengembangkan kampanye iklan yang tersegmentasi untuk menarik minat konsumen senior. Hal ini dapat melibatkan penawaran produk atau jasa yang sesuai dengan kebutuhan dan preferensi mereka.
- Segmentasi Berdasarkan Level Pendapatan : Melihat ketertarikan yang tinggi dari pengguna dengan pendapatan rendah terhadap iklan, kita dapat mengoptimalkan anggaran iklan dengan menargetkan segmen pasar ini. Dengan menawarkan solusi hemat biaya, kita dapat membangun loyalitas pelanggan jangka panjang.

Business Simulation

Tanpa Menggunakan Machine Learning

Asumsi

Biaya per iklan	Rp 1.000
Nilai rata-rata pembelian	Rp 500.000
Conversion rate (CR)	5%
Target jumlah pengiklanan	1.000 pengguna
Jumlah pengguna benar-benar klik iklan	500 orang
Click rate	500/1.000 * 100 = **50%**

Perhitungan

Biaya iklan	Rp 1.000 x 1.000 pengguna = Rp 1.000.000
Jumlah pembelian :	Jumlah pengguna benar-benar klik iklan x CR = 500 pengguna x 5% = 25 pembelian
Revenue	Jumlah pembelian x Nilai rata-rata pembelian = 25 pembelian x Rp 500.000 = Rp 12.500.000
Profit	Revenue - Biaya iklan = Rp 12.500.000 - Rp 1.000.000 = **Rp 11.500.000**

Dengan Menggunakan Machine Learning

Asumsi

Biaya per iklan	Rp 1.000
Nilai rata-rata pembelian	Rp 500.000
Conversion rate (CR)	5%
Target jumlah pengiklanan	1.000 pengguna
Jumlah pengguna benar-benar klik iklan	Precision x Target jumlah pengguna = 98% * 1.000 pengguna = 980 pengguna
Click rate	980/1.000 * 100 = **98%**

Perhitungan

Biaya iklan	Rp 1.000 x 1.000 pengguna = Rp 1.000.000
Jumlah pembelian :	Jumlah pengguna benar-benar klik iklan x CR = 980 pengguna x 5% = 49 pembelian
Revenue	Jumlah pembelian x Nilai rata-rata pembelian = 49 pembelian x Rp 500.000 = Rp 24.500.000
Profit	Revenue - Biaya iklan = Rp 24.500.000 - Rp 1.000.000 = **Rp 23.500.000**

	Tanpa Machine Learning	Dengan Machine Learning
Click Rate	50%	98%
Profit	Rp 11.500.000	Rp 23.500.000

Kesimpulan

Dengan membandingkan profit dan click rate sebelum dan sesudah penerapan model, kita dapat melihat bahwa dengan penerapan model, click rate meningkat dari **50%** menjadi **98%**, dan profit juga meningkat dari **Rp 11.500.000** menjadi **Rp 23.500.000** (peningkatan sebesar **104,35%**).

Terimakasih