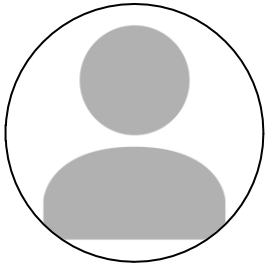
Predict Clicked Ads Customer Classification by using Machine Learning

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



Created by:
Ferry Irwanto
ferryirwanto89@gmail.com
linkedin.com/in/ferryirwanto

"Data Analyst with hands-on experience in financial risk analysis, KPI tracking, and dashboard development. Proficient in SQL, Python, and BI tools to transform raw data into actionable insights. Skilled at managing multiple projects independently, delivering accurate business solutions under tight deadlines and improving operational efficiency."

Untuk selengkapnya, dapat melihat jupyter notebook disini

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 ”

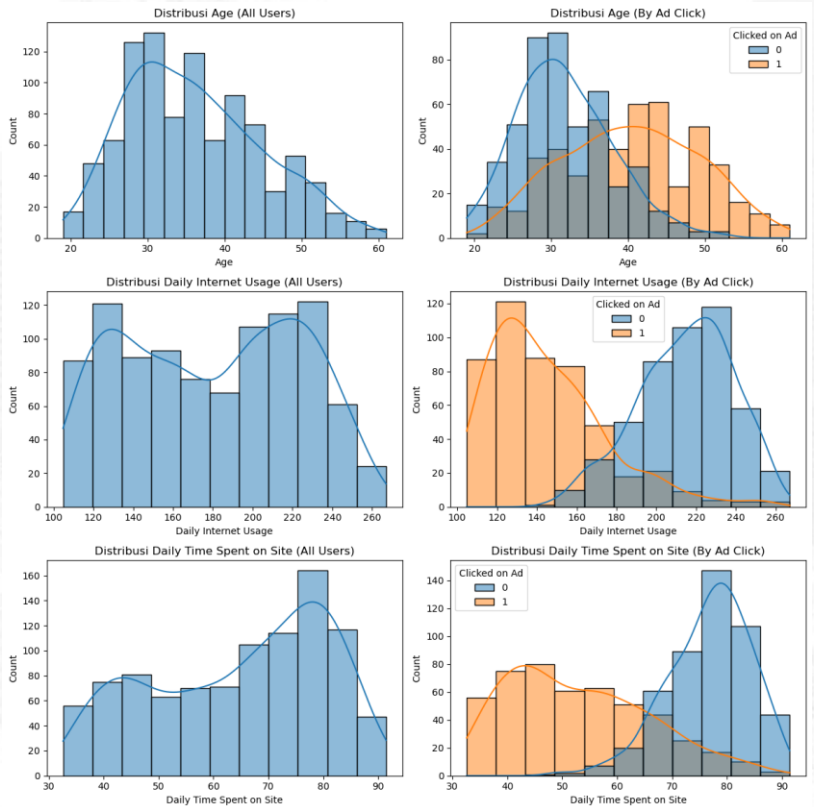
Statistical Analysis

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Clicked on Ad
count	963.000000	963.000000	9.630000e+02	963.000000	963.000000
mean	64.829200	36.049844	3.855628e+08	179.716106	0.503634
std	15.892075	8.764154	9.380390e+07	43.867289	0.500247
min	32.600000	19.000000	9.797550e+07	104.780000	0.000000
25%	50.600000	29.000000	3.296658e+08	138.615000	0.000000
50%	68.010000	35.000000	3.991039e+08	182.200000	1.000000
75%	78.365000	42.000000	4.591870e+08	218.550000	1.000000
max	91.430000	61.000000	5.563936e+08	267.010000	1.000000

	Male	Timestamp	city	province	category
count	963	963	963	963	963
unique	2	960	30	16	10
top	Perempuan	5/20/2016 12:17	Bandung	Daerah Khusus Ibukota Jakarta	Otomotif
freq	502	2	64	244	108

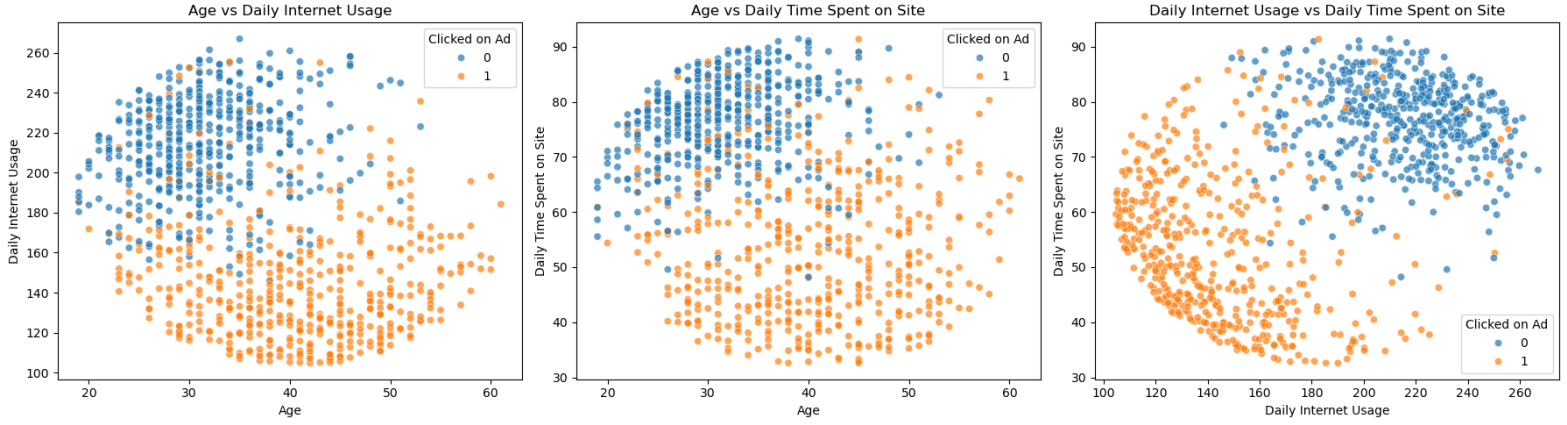
- Dataset seimbang: jumlah user yang klik iklan dan tidak klik hampir sama.
- Rata-rata usia user: 36 tahun, mayoritas berada di usia produktif.
- Gender relatif seimbang, kategori iklan didominasi otomotif, dan sebagian besar user berasal dari DKI Jakarta.

Univariate Analysis



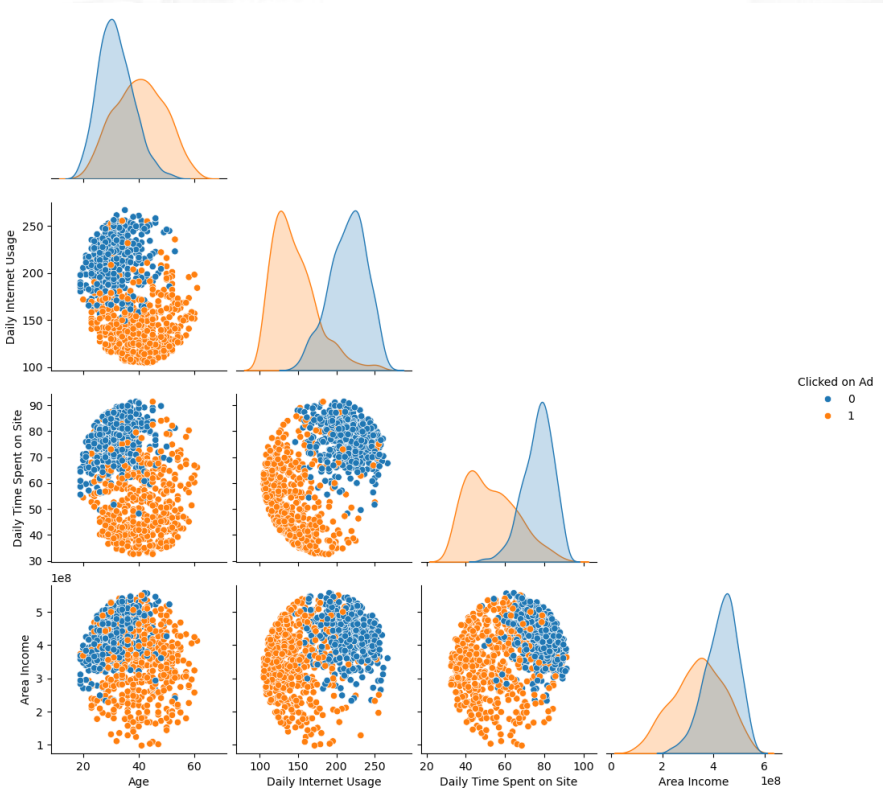
- User tua lebih sering klik iklan.
- Internet usage tinggi → jarang klik.
- Time on site lama → jarang klik.

Bivariate Analysis



- Umur vs Internet Usage → User tua dengan internet usage rendah cenderung klik.
- Umur vs Time on Site → User tua dengan waktu singkat di site lebih sering klik.
- Internet Usage vs Time on Site → Ada dua cluster jelas: klik (rendah-rendah), tidak klik (tinggi-tinggi).

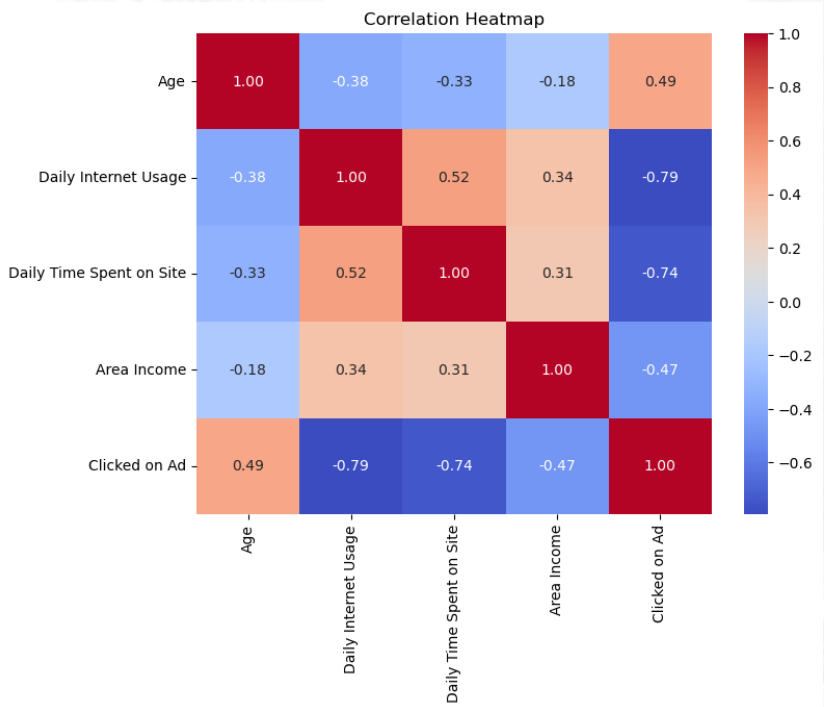
Multivariate Analysis



Kombinasi variabel menunjukkan pola cluster:

- Klik → umur lebih tua, internet usage rendah, site time singkat.
- Tidak klik → umur muda, internet usage tinggi, site time lama.

Heatmap Analysis



- Age positif dengan klik iklan (0.49).
- Daily Internet Usage negatif kuat dengan klik iklan (-0.79).
- Daily Time Spent on Site negatif kuat dengan klik iklan (-0.74).

Insight Praktis:

1. Iklan lebih efektif ditargetkan pada user **lebih tua dengan aktivitas online moderat**.
2. User muda dengan aktivitas online tinggi cenderung **mengabaikan iklan**.
3. Segmentasi berbasis usia & perilaku digital dapat meningkatkan **efektivitas dan efisiensi pemasaran**.

Null & Duplicate handling

```
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Daily Time Spent on Site  987 non-null    float64
 1   Age                     1000 non-null   int64
 2   Area Income             987 non-null    float64
 3   Daily Internet Usage    989 non-null    float64
 4   Male                    997 non-null    object
```

```
df_clean = df.copy()
df_clean = df_clean.dropna()
df_clean = df_clean.drop_duplicates()
df_clean = df_clean.rename(columns={"Male": "Gender"})
df_clean.info()
```

✓ 0.0s

```
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Daily Time Spent on Site  963 non-null    float64
 1   Age                     963 non-null    int64
 2   Area Income             963 non-null    float64
```

Beberapa kolom memiliki null value, dan ketika dilihat memang tidak berjumlah banyak, sekitar 1.3%.

Melihat nilai tersebut, maka melakukan penghapusan data tidak masalah. Terlebih lagi total data yang terhapus masih <5% sehingga tidak akan berpengaruh besar pada data yang digunakan.

Encoding, timestamp, and splitting

```
df_clean["Clicked on Ad"] = df_clean["Clicked on Ad"].map({"Yes": 1, "No": 0})
df_clean = pd.get_dummies(df_clean, columns=["Gender", "city", "province", "category"], drop_first=True)
```

✓ 0.0s

```
df_clean["Timestamp"] = pd.to_datetime(df_clean["Timestamp"], errors="coerce")
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["DayOfWeek"] = df_clean["Timestamp"].dt.dayofweek
```

✓ 0.0s

```
X = df_clean.drop(columns=["Clicked on Ad", "Timestamp"])
y = df_clean["Clicked on Ad"]
```

✓ 0.0s

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42, shuffle=True)
print("X_train and X_test size:", X_train.shape, X_test.shape)
print("y_train and y_test size:", y_train.shape, y_test.shape)
```

✓ 0.0s

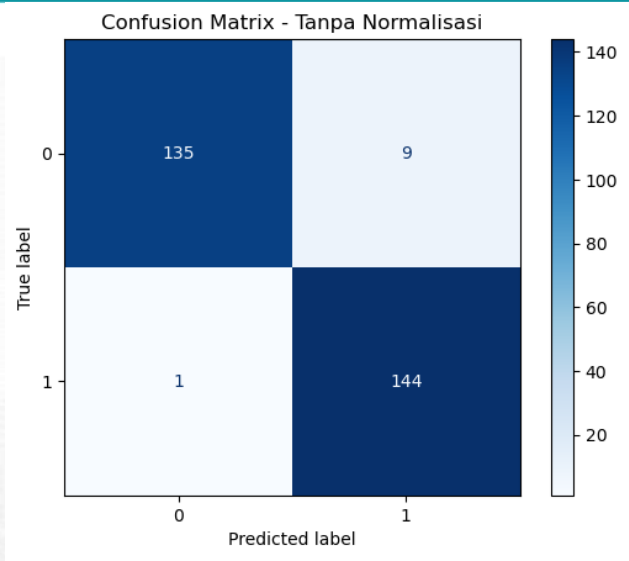
```
X_train and X_test size: (674, 63) (289, 63)
y_train and y_test size: (674,) (289,)
```

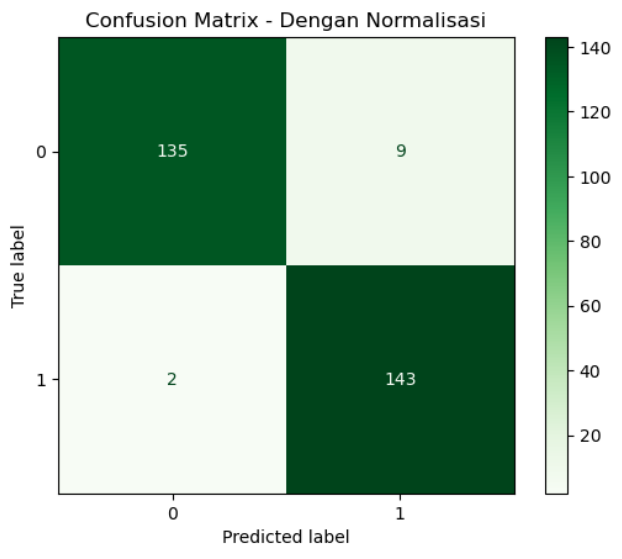
- Untuk data categorical, akan dilakukan one-hot encoding untuk mempermudah model memahami struktur datanya
- Ekstraksi timeframe dilakukan dengan function dt.{time}
- Splitting data dilakukan sebesar 70/30 karena datanya terbatas (963) dan perlu nilai yang balance untuk model belajar dan evaluasi

Data Modeling

- Model yang dipakai berupa Random Forest. Hal ini dilakukan karena:
  - RF tidak sensitif terhadap normalisasi
  - Mampu menangani data campuran (numerik + kategorikal) termasuk encoded data
  - RF dapat mencegah overfit karena sistem *Bagging*
  - Punya performa baik dengan dataset yang tidak terlalu banyak (~1000)

Komparasi Confusion Matrix dan Hasil Evaluasi





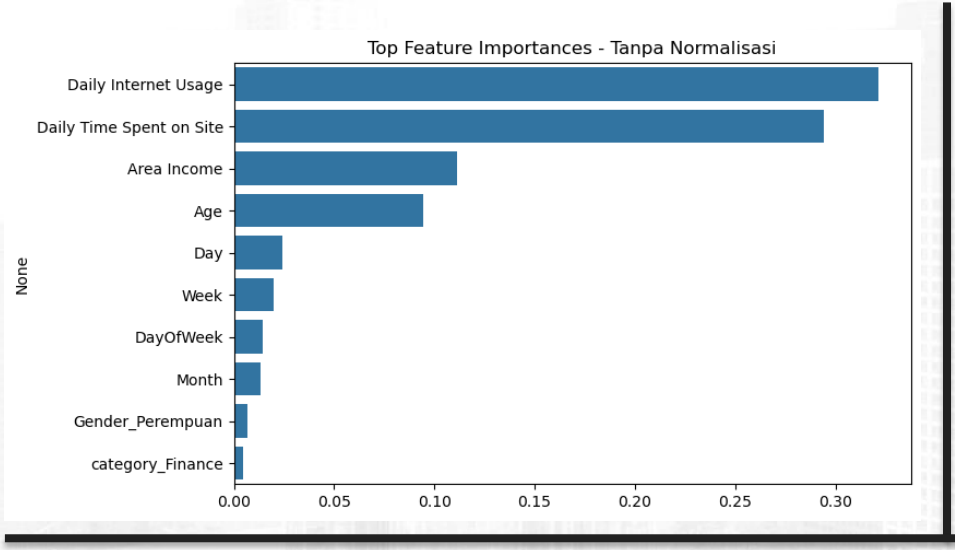
Classification Report - Tanpa Normalisasi

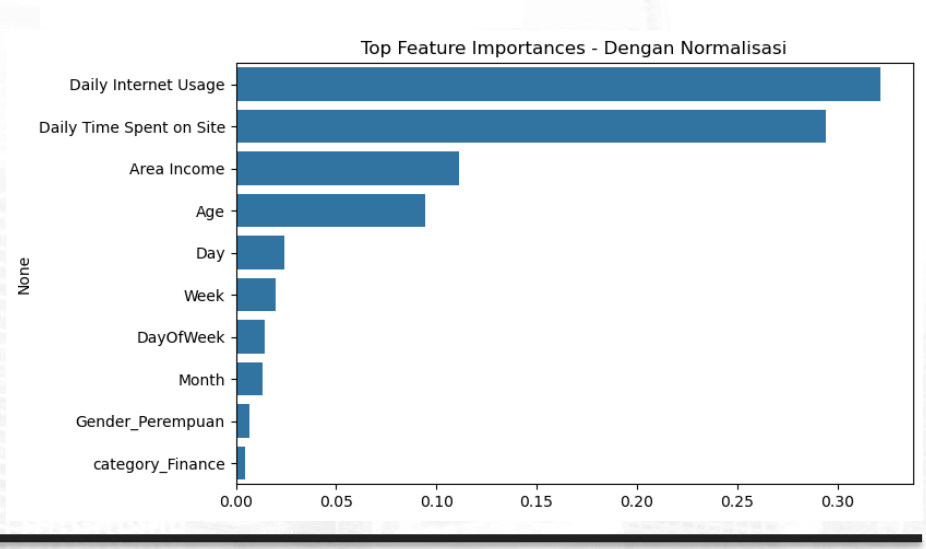
	precision	recall	f1-score	support
0	0.99	0.94	0.96	144
1	0.94	0.99	0.97	145
accuracy			0.97	289
macro avg	0.97	0.97	0.97	289
weighted avg	0.97	0.97	0.97	289

Classification Report - Dengan Normalisasi

	precision	recall	f1-score	support
0	0.99	0.94	0.96	144
1	0.94	0.99	0.96	145
accuracy			0.96	289
macro avg	0.96	0.96	0.96	289
weighted avg	0.96	0.96	0.96	289

Komparasi Feature Importance





Hasil Interpretasi:

- Tanpa normalisasi sedikit lebih baik (97% vs 96%).
- Dari grafik feature importance, dua fitur teratas yang paling berpengaruh dalam prediksi adalah **Daily Internet Usage** dan **Daily Time Spent on Site**
- Keberhasilan marketing iklan sangat dipengaruhi oleh kebiasaan penggunaan internet dan durasi kunjungan ke website.
- User dengan internet usage rendah & site time singkat → lebih tinggi kemungkinan klik iklan.
- Segmentasi berdasarkan dua variabel ini bisa jadi dasar untuk targeting iklan yang lebih efektif.

Rekomendasi Bisnis

- Dari hasil EDA dan feature importance, variabel utama yang menentukan klik iklan adalah:
  - Daily Internet Usage (penggunaan internet harian)
  - Daily Time Spent on Site (waktu yang dihabiskan di website)

- Insight:
  - User dengan internet usage rendah dan waktu kunjungan singkat lebih tinggi kemungkinan klik iklan.
  - User muda dengan internet usage tinggi & waktu lama di site cenderung mengabaikan iklan.

- Rekomendasi Bisnis:
  - Fokus targeting iklan pada segmen pengguna dengan internet usage rendah & site time singkat.
  - Batasi distribusi iklan ke heavy internet users untuk efisiensi biaya.

Simulasi Bisnis

```python
# Asumsi simulasi
baseline_conv_rate = df_clean["Clicked on Ad"].mean()
total_users = 1000
marketing_cost_per_user = 1000
revenue_per_conversion = 20000
ml_precision_raw = 0.94
```

```python
def simulate(name, conv_rate, total_users, cost_per_user, revenue_per_conversion):
    cost = total_users * cost_per_user
    conversions = total_users * conv_rate
    revenue = conversions * revenue_per_conversion
    profit = revenue - cost
    roi = (profit / cost) if cost != 0 else None
    return {
        "scenario": name,
        "total_users_targeted": total_users,
        "conversion_rate": conv_rate,
        "expected_conversions": conversions,
        "total_cost": cost,
        "total_revenue": revenue,
        "total_profit": profit,
        "roi": roi
    }
```

- **Targeting dengan ML hampir menggandakan profit** dibandingkan strategi random (Rp 9 juta → Rp 17,8 juta).
- Conversion rate melonjak dari **50% menjadi 94%**, artinya iklan lebih tepat sasaran.
- ROI meningkat signifikan → investasi pemasaran menjadi jauh lebih efisien dengan bantuan model ML.
- Faktor penentu keberhasilan adalah segmentasi user berbasis **Daily Internet Usage** dan **Daily Time Spent on Site**, sesuai hasil feature importance.

	scenario	total_users_targeted	conversion_rate	expected_conversions	total_cost	total_revenue	total_profit	roi
0	No ML (baseline - target all users randomly)	1000	0.503634	503.6	Rp 1,000,000	Rp 10,072,690	Rp 9,072,690	9.07
1	With ML (use model precision as conversion rate)	1000	0.940000	940.0	Rp 1,000,000	Rp 18,800,000	Rp 17,800,000	17.80