# Predict Customer Clicked Ads Classification by Using Machine Learning


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


Rakamin Academy


**Created by:**
**Moh. Harwin Prayoga**
mh.prayoga11@gmail.com
https://www.linkedin.com/in/mhprayoga/


**“An enthusiastic learner, analytical, and flexible graduate of bachelor's degree of Engineering Physics at Institut Teknologi Sepuluh Nopember. I had experience in leadership and teamwork in various organizations and events. Moreover, I have a decent ability in English and operating various data programming software such as MS Excel, Python, SQL, etc. I am excited about seeking a challenge in the field of data where my passion, education, and training background can be fully utilized. ”**

# Overview

Rakamin Academy

"A company in Indonesia wants to know the effectiveness of an advertisement that they display, this is important for the company to be able to find out how much the advertising has been marketed so that it can attract customers to see the advertisement.

By processing historical advertisement data and finding insights and patterns that occur, it can help companies determine marketing targets. The focus of this case is to create a machine learning classification model to determine the right target customers."

# Data Exploration

```
<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
```

- The dataset has 1000 rows and 11 columns.
- Unnamed: 0 column is identified as ID that can be dropped or renamed if necessary.
- Contains 3 data types: float64, int64, object.
- Timestamp column data type will be changed to datetime instead of object.
- Male column contains 2 unique values, which will be renamed to Gender
- There are 4 columns containing null values.

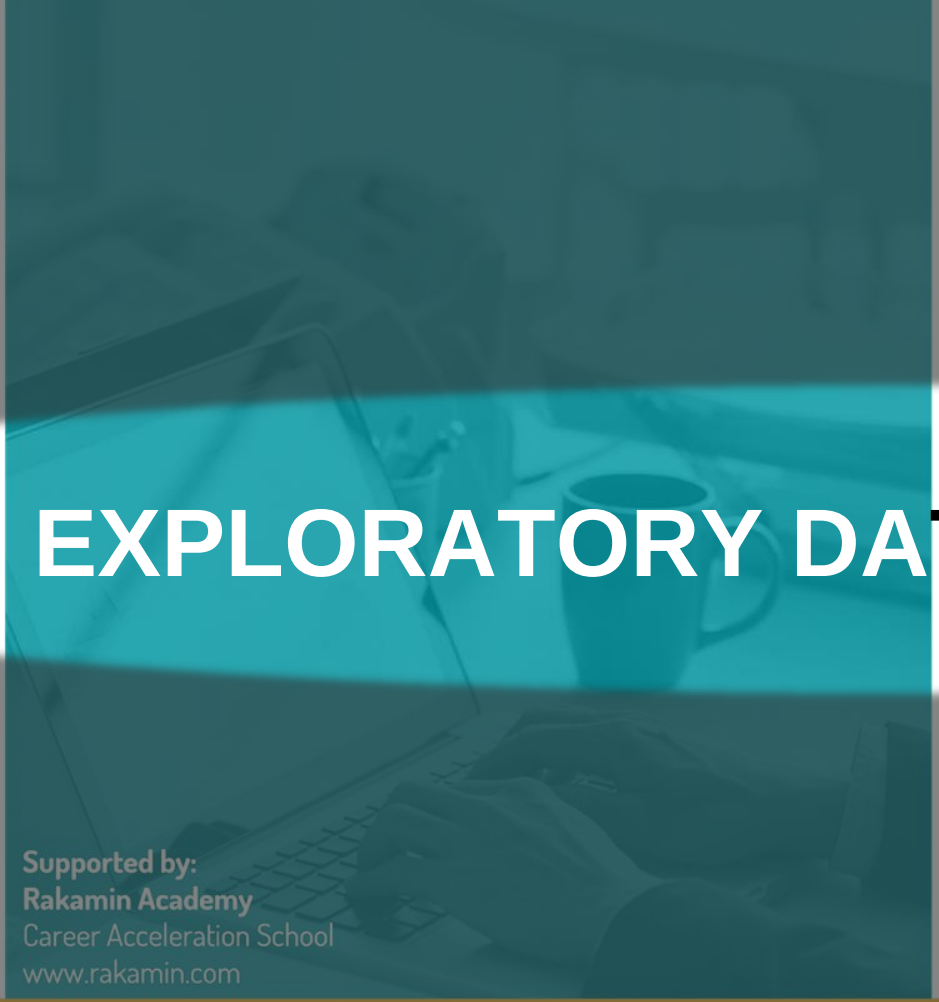
For details, check jupyter notebook here

# Feature Engineering

| No | Feature | Explanation |
|---|---|---|
| 1 | Timestamp | Convert data type to datetime |
| 2 | Unnamed: 0 | Renamed to ID |
| 3 | Male | Renamed to Gender |

For details, check jupyter notebook here

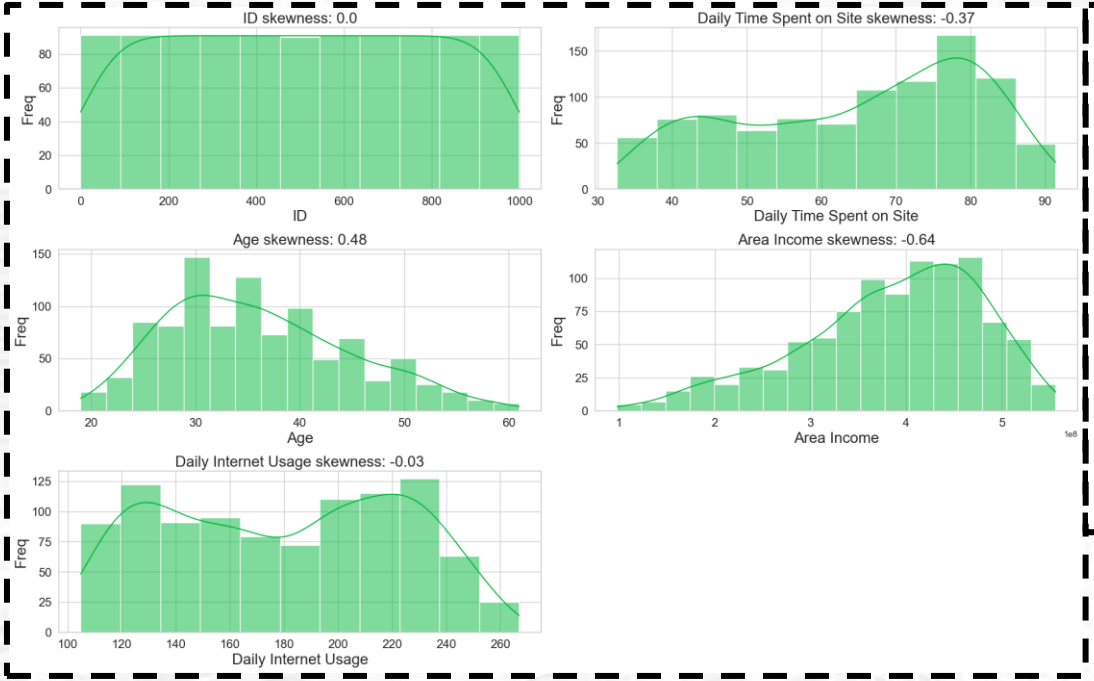
# EXPLORATORY DATA ANALYSIS (EDA)

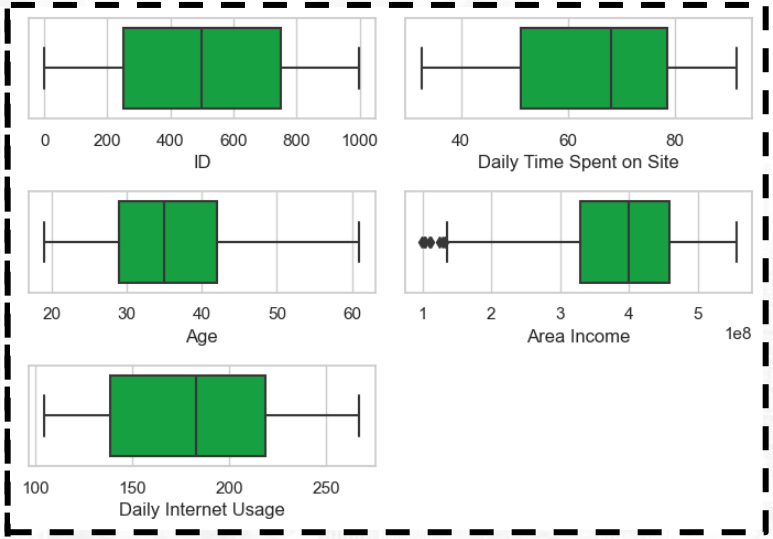


Supported by:
Rakamin Academy
Career Acceleration School
www.rakamin.com

# Univariate Analysis

## Numerical:



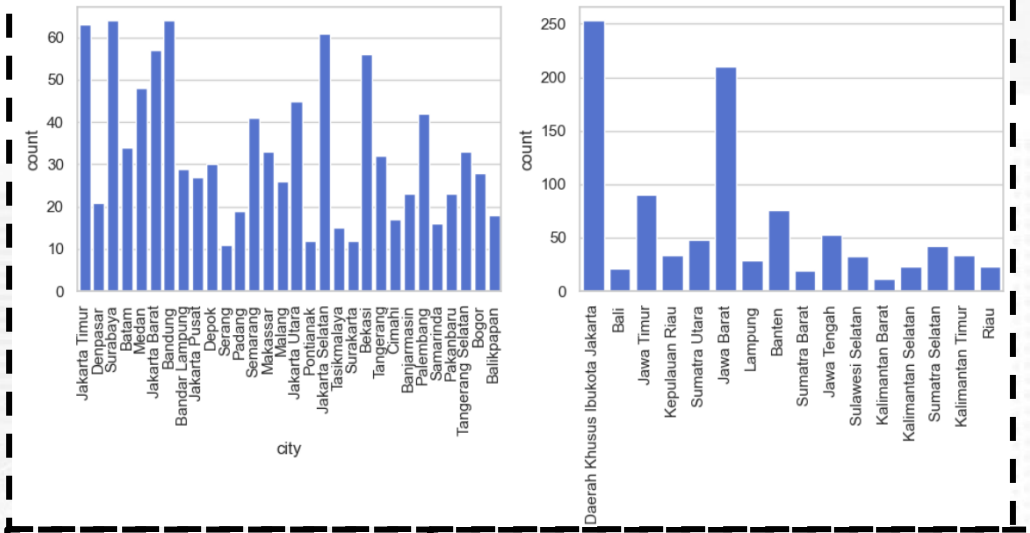



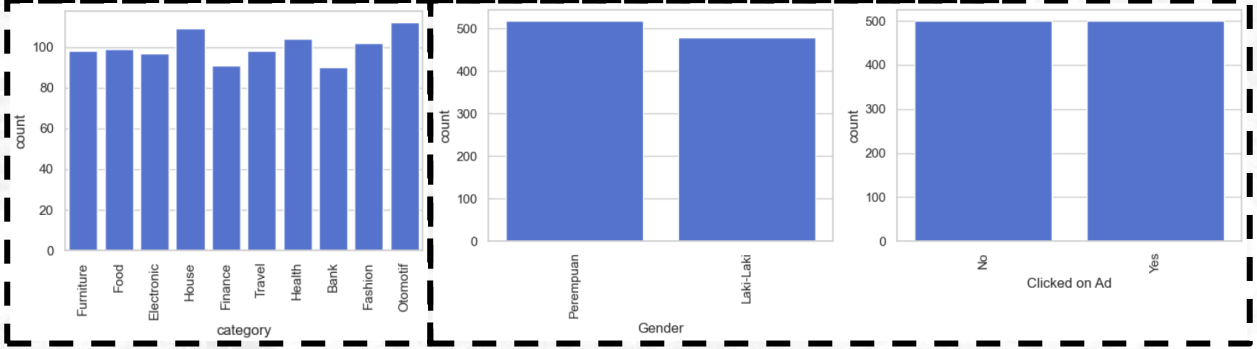

Almost all numerical columns seem to have symmetrical(normal) distribution, except Income column (left skewed).

For details, check jupyter notebook here

# Univariate Analysis

## Categorical:



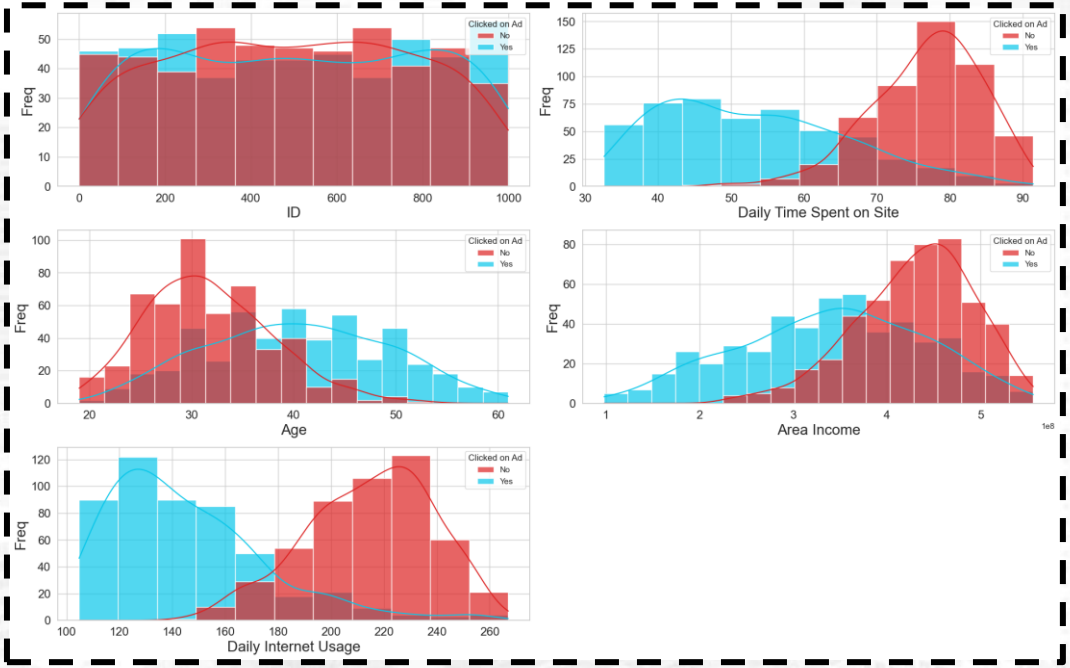




- In Gender column has almost balance value between "Laki-laki" or "Perempuan".
- Clicked on Ad column has balanced value.
- province column has 2 dominant value, that is "Daerah Khusus Ibukota Jakarta" and "Jawa Barat".
- Category column has most likely balanced value.

For details, check jupyter notebook here

# Bivariate Analysis

## Numerical:

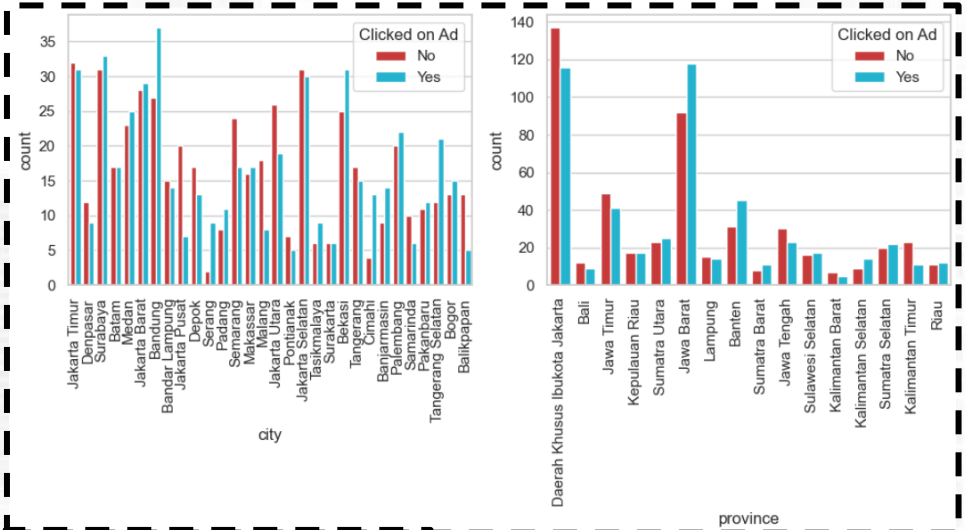


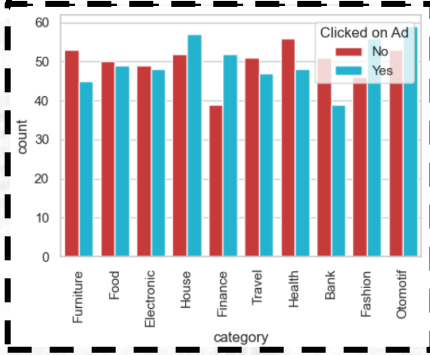

- ID column has a balance distribution for "Yes" and "No".
- The longer time spent on site, the more probability of customers won't click the ads. And vice versa.
- The average Age of customers that clicked the ads is 40 years old and not clicked the ads are 31 years old.
- The more Income that customers get, the more probability of customers won't click the ads. And vice versa.
- Customers that have not clicked the ads have more Daily Internet Usage around 220 minutes per day, however, the customers who clicked the ads have around 140 per day.

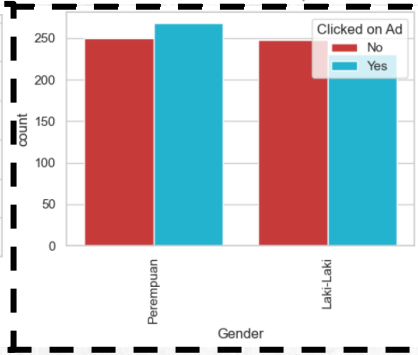
For details, check jupyter notebook here

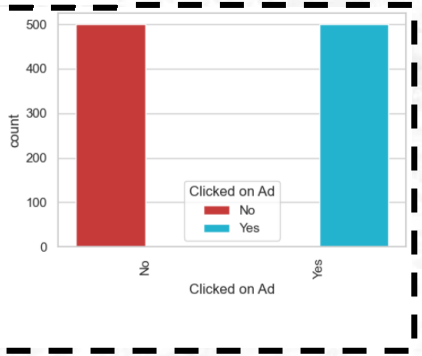
# Bivariate Analysis

## Categorical:



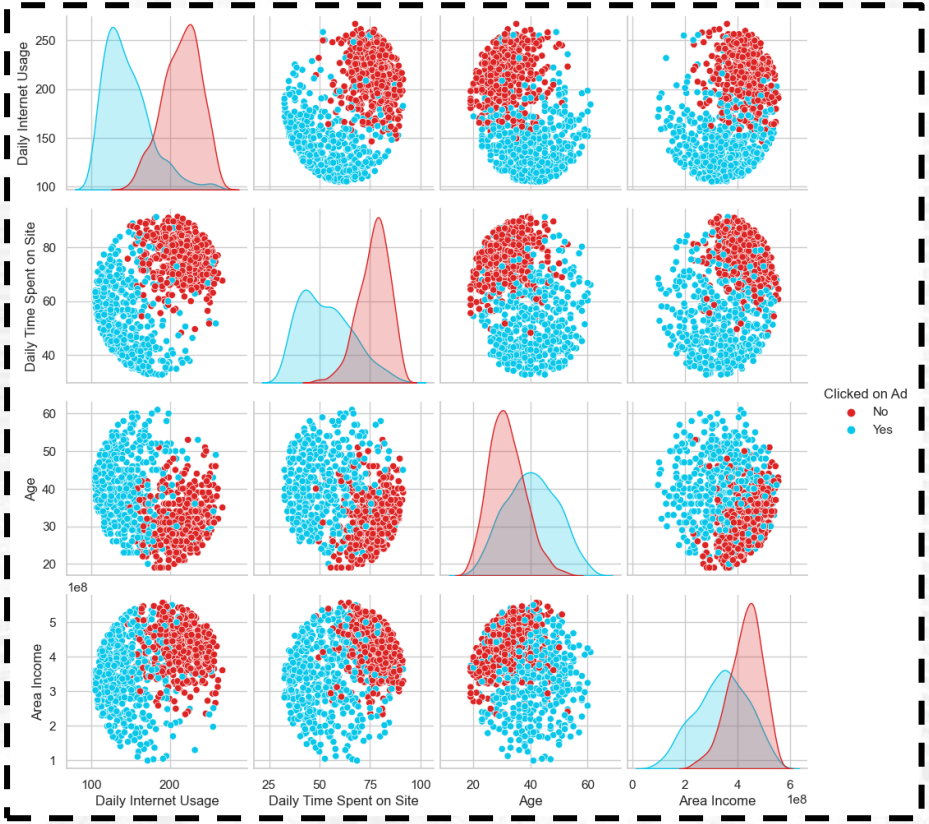




- "Perempuan" tends to do more clicked on ads than "Laki-laki"
- The highest amount of clicked on ads by city was in Bandung, Surabaya, and Bekasi.
- The top 3 highest amount of clicked on ads are "Daerah Khusus Ibukota Jakarta", "Jawa Barat", dan "Banten"
- The top 3 highest amount of customers who clicked on ads are "Otomotif", "House", and Fashion. Meanwhile "Health" category has the most customers who not clicked on ads.

For details, check jupyter notebook here

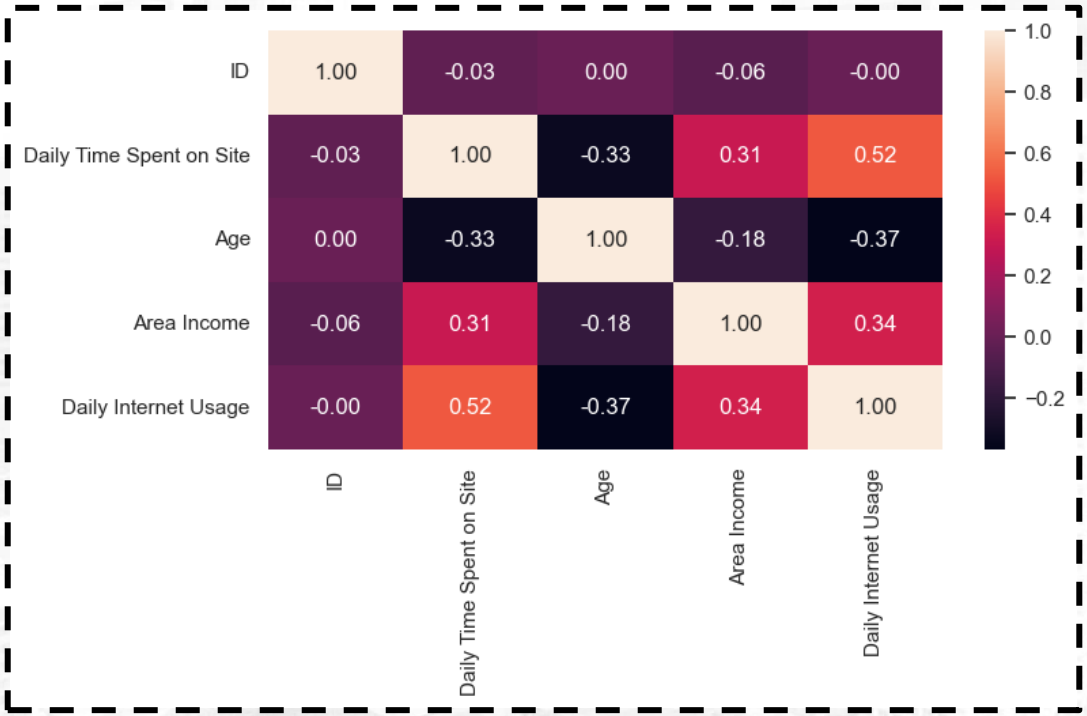
# Multivariate Analysis




- The older Age and the more Daily Time Spent on Site and Daily Internet Usage of customers to click on ads.
- The lower Daily Internet Usage and Daily Time Spent on Site, the more users tend to click on ads.
- The lower customers' income, the higher Daily Internet Usage and Daily Time Spent on Site.

For details, check jupyter notebook here

# Multivariate Analysis




- **Daily Time Spent on Site** has **strong positive** correlation with **Daily Internet Usage**, **medium positive** correlation with Area Income, and **medium negative** correlation with **Age**.
- **Age** has **medium negative** correlation with **Daily Internet Usage** and **Daily Time Spent on Site**.
- **Area Income** has **medium positive** correlation with **Daily Internet Usage** and **Daily Time Spent on Site**.
- **Daily Internet Usage** has **positive strong** correlation with **Daily Time Spent on Site**, **medium negative** correlation with **Age**, and **medium positive** correlation with **Area Income**.

For details, check jupyter notebook here

# DATA CLEANING & PROCESSING

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

# Data Cleaning & Processing

Rakamin Academy

**Handling Missing Values**

- Input **“Area Income”** with **Median**
- Input **“Daily Time Spent on Site”** & **“Daily Internet Usage”** with **Mean**
- Input **“Male”** with **Mode**

**Handling Duplicated Data**

**No duplicated data**

**Feature Engineering**

- **Extracting** **`Timestamp`** column to get **'Date’, `Time’, 'Week’, 'Weekday’, 'Year`, `Month`, and `Day`** columns
- **Replacing** value **“'Daerah Khusus Ibukota Jakarta”** to **“DKI Jakarta”**

**Handling Outlier**

Using **IQR Method** to handle outliers

**Feature Encoding**

- Using **LabelEncoding** to **“Male”** and **“Clicked on Ad”**
- Using **One Hot Encoding** to **“category”**

For details, check jupyter notebook here

# Data Cleaning & Processing

Rakamin Academy

**Drop Unnecessary Columns**

- Dropping **Unnamed: 0** because has **too many unique values**.
- Dropping **Timestamp** column because values **had been extracted** to **Date**, **Time**, **Week**, **Weekday**, **Year**, **Month**, and **Day**.
- Dropping **Year** column because only has **1 unique** value.

**Handling Duplicated Data**

Filtering to only **'float64', 'int64','uint8'** data types

**Splitting Feature & Target**

- Dividing **Features** and **target (Clicked on Ad)** column

**Feature Scaling**

**Data Standardization** with **StandardScaler**

**Splitting to Train set and Test set**

- Data will be **splitted to 80% data train and 20% data test**

For details, check jupyter notebook here

# DATA MODELING

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

# Data Modeling (Before Standardization)

## Experiment 1 Modeling with Hyperparameter Tuning

| model | Recall_test | Recall_train | Accuracy_test | Accuracy_train | Time_Elapsed |
|---|---|---|---|---|---|
| Logistic Regression | 0.908 | 0.874 | 0.949 | 0.907 | 4.990 |
| KNearest Neighbors | 0.586 | 0.574 | 0.728 | 0.700 | 198.470 |
| Decision Tree | 0.851 | 0.945 | 0.903 | 0.951 | 3.990 |
| Random Forest | 0.943 | 0.982 | 0.949 | 0.986 | 61.830 |
| Gradient Boosting | 0.943 | 0.977 | 0.949 | 0.982 | 3.990 |

- ✓ Almost all models don't have wide gaps between train and test.
- ✓ **KNN** has the most **Time_Elapsed** and **lowest score**.
- ✓ **Decision Tree** and **Gradient Boosting** have the lowest **Time_Elapsed**.
- ✓ **Random Forest** has the overall best score for both **Accuracy** and **Recall**. But a bit **high in time required.**

For details, check jupyter notebook here

# Data Modeling (After Standardization)

## Experiment 2 Modeling with Hyperparameter Tuning

| model | Recall_test | Recall_train | Accuracy_test | Accuracy_train | Time_Elapsed |
|---|---|---|---|---|---|
| Logistic Regression | 0.931 | 0.935 | 0.964 | 0.963 | 4.000 |
| KNearest Neighbors | 0.874 | 0.889 | 0.944 | 0.940 | 258.300 |
| Decision Tree | 0.931 | 0.962 | 0.949 | 0.970 | 1.990 |
| Random Forest | 0.943 | 0.985 | 0.949 | 0.988 | 65.820 |
| Gradient Boosting | 0.954 | 0.985 | 0.954 | 0.987 | 3.990 |

- ✓ **KNN** has the highest number of **Time Elapsed**.
- ✓ **Random Forest** has the best score in **Recall**.
- ✓ **Gradient Boosting** has the slight better score in **Accuracy**.
- ✓ **Logistic Regression**, **Decision Tree**, and **Gradient Boosting** has similar **Time Elapsed** score to each other and the top 3 **fastest time** required.

For details, check jupyter notebook here

# Data Modeling (Conclusion)

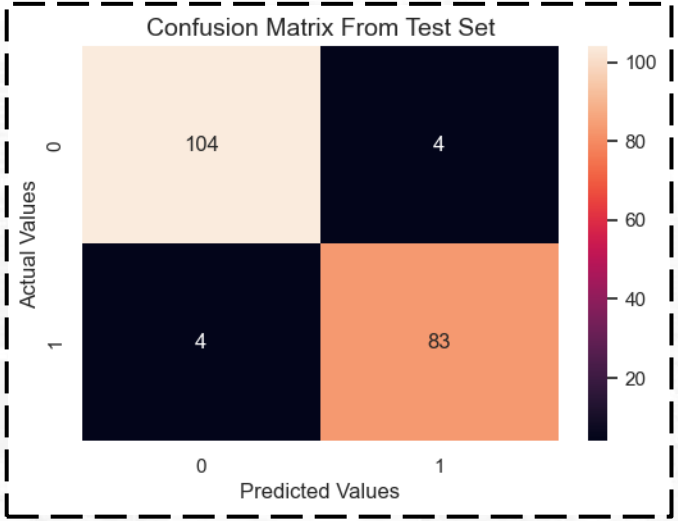
Rakamin Academy

“ By considering both Recall, Accuracy, and Time_Elapsed score. **Gradient Boosting with Standardization / Normalization** is chosen for decent score and fast time required. “

For details, check jupyter notebook here

# Selected Model

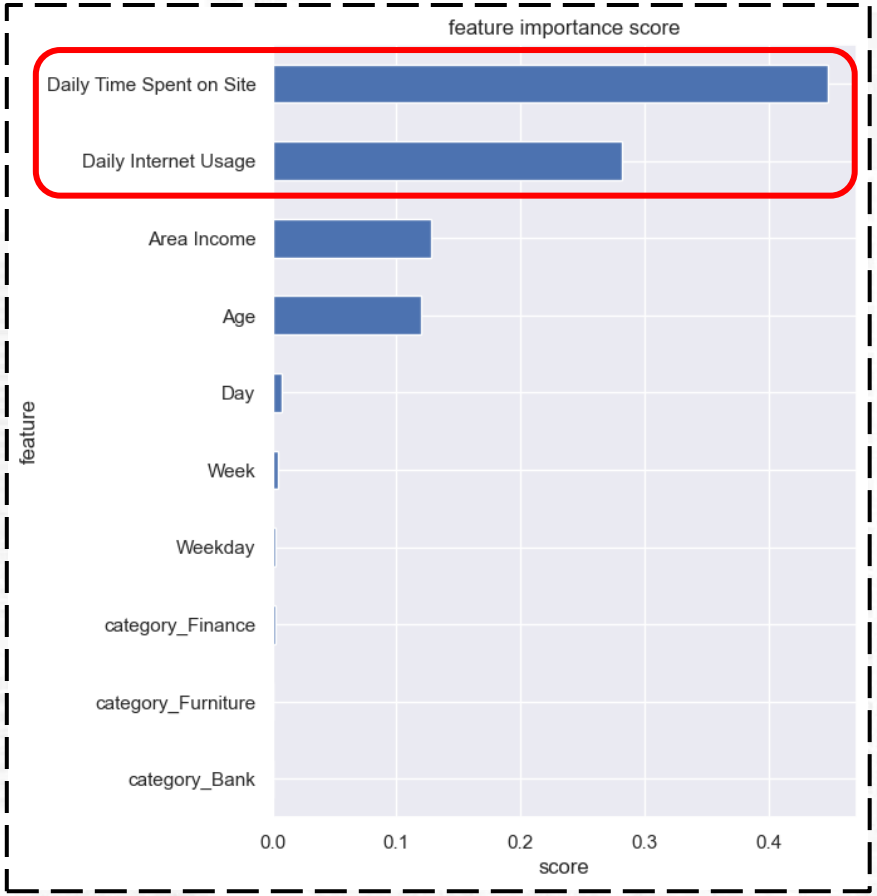
```
grad_best2 = GradientBoostingClassifier(n_estimators = 50,
                                        criterion = 'friedman_mse',
                                        max_depth = 3,
                                        min_samples_split = 3,
                                        min_samples_leaf = 1,
                                        max_features = 'sqrt',
                                        loss = 'exponential').fit(X2_train, y2_train)

y_pred_train_best = grad_best2.predict(X2_train)
y_pred_best = grad_best2.predict(X2_test)

eval_classification(grad_best2, y_pred_train_best, y_pred_best, y2_train, y2_test)
```




**As a result of the confusion matrix shown above for both data train and data test, Gradient Boosting was appropriate to apply to the existing dataset**

For details, check jupyter notebook here

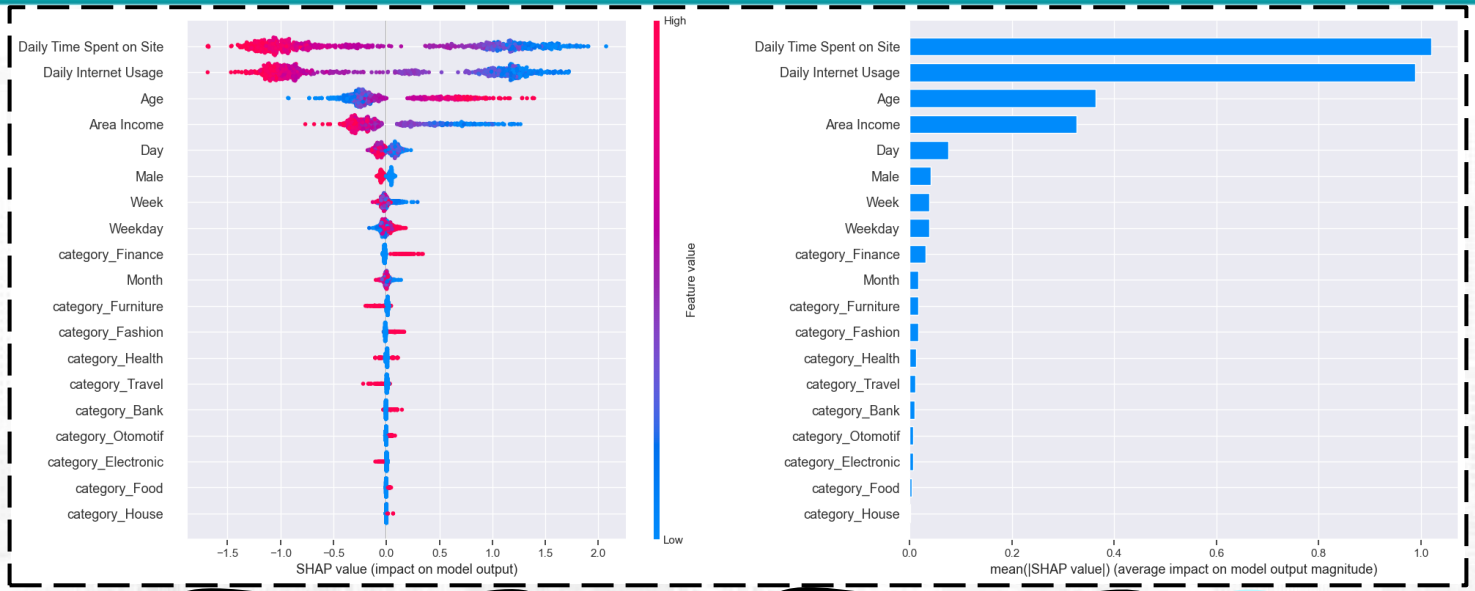
# Feature Importance




**Daily Time Spent on Site** is the **most important** feature of this model and following by **Daily Internet Usage**

For details, check jupyter notebook here

# SHAP Observation




" The two most important features that affect to user whether to click on ad or not are **Daily Internet Usage** and **Daily Time Spent on Site.** "

- The **fewer** the **Daily Internet Usage**, the **more** users click on ad. Otherwise, the **higher** the number of **Daily Internet Usage** the **fewer** users will click on ad.
- The **lower** the number of **Daily Time Spent on Site** the **more** users click on ad. Otherwise, the **higher** the number of **Daily Time Spent on Site** the **fewer** users will click on ad.

For details, check jupyter notebook here

# Business Recommendation & Simulation

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

# Business Recommendation based on EDA & Feature

According to insights that we have gathered on EDA and Feature Importance, we can determine the proper recommendation to increase customers to click on ads:

1. Customers who do not click on ads tend to be younger than the customers who do click on ads. That means the younger customers are not easily distracted by ads than the older customers even though they have spent more time on site. Hence**, the ads should be ads that most probably relate to what customers searched for and not take much time and space on UI display**.

2. **Displaying ads to customers who have an income of more than 4 million is not recommended**. That's probably because those customers when browsing the company website focus on what they are visiting for, so they are not too interested in other options that distract them.

3. The company has various sale categories. So that we can **choose a specific category on a specific customer base on the trends of each category**.

For details, check jupyter notebook here

# Business Simulation

After a model is created, then we can create a simulation when the model is applied.

## Without Applying ML Model

```
Clicked on Ad has 2 with data dtypes: object
No     500
Yes    500
Name: Clicked on Ad, dtype: int64
```

**Assumpting :**

- Budget per advertisement is IDR 1.000
- revenue per click IDR 4.000

- Calculating Cost:

cost = ads_cost * n_customer
cost = 1.000 * 1.000
cost = 1.000.000

- The conversion rate was 50% because 500 of 1000 users clicked on ads.
- Hence profit will be:

profit = (clicked_cust * revenue) - cost
profit = (500 * 4.000) - 1.000.000

profit = **IDR 1.000.000.** ← **Profit 1 : 1 with cost**

For details, check jupyter notebook here

# Business Simulation

## With Applying ML Model

According to ML Model performance, we can get 96% accuracy. So that means assuming we have 1000 customers, we can calculate the profit that can potentially get:

**Assumpting :**

- Budget per advertisement is IDR 1.000
- revenue per click IDR 4.000

The same total cost of 1000 customers: 1.000.000

profit = (clicked_cust * revenue) - cost

profit = (960 * 4.000) - 1.000.000

profit = **IDR 2.840.000**

**Profit 2,84 : 1 with cost**

For details, check jupyter notebook here

# Business Simulation

Rakamin Academy

## CONCLUSION

“Hence comparing with before and after using ML, we potentially significantly increase the revenue so **we can get 284% more profit**.“

For details, check jupyter notebook here