Predict Clicked Ads Customer Classification by using Machine Learning

Created by:
Saila Fikriyya
sailafs@gmail.com
linkedin.com/in/sailafs1203/

I'm a health psychology graduate turned data scientist, bridging the gap between human behavior and data-driven solutions. With a Master's in Health Psychology from the University of Stirling and ongoing training in data science, I combine expertise in research methods, data analysis, and understanding human behavior. My experience spans from studying post-traumatic growth in cancer survivors to applying data science techniques in digital healthcare and career development. I'm passionate about leveraging interdisciplinary skills to create evidence-based strategies that improve health outcomes and overall well-being, always prioritizing a human-centric approach in collaborative environments.

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

Overview

Rakamin Academy

"A company in Indonesia wants to know the effectiveness of an advertisement they launched online. This is important for the company to understand how well the advertisement reaches its target audience, so **they can attract customers to click on the ad**.

By processing historical advertisement data and uncovering insights and patterns, it can assist companies in determining their marketing targets. The focus of this case is to create a machine learning classification model that serves to identify the right target customers. "

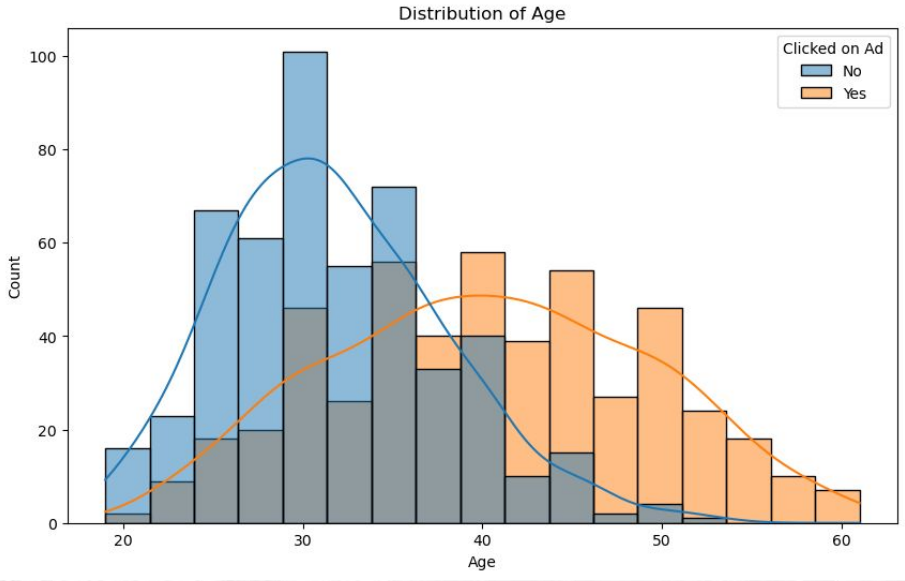
Statistical Analysis

- In terms of categorical features over advertisement, no significant difference observed.
  - Gender (p=.296)
  - Category (p=.695)
  - City (p=0.206)
  - Province (p=0.381)
- In terms of numerical features, age is having significant difference over advertisement (p=.000). However, other numerical features could not be tested due to missing data which will be addressed in the next stage.

For more information, see jupyter notebook here

EDA: Univariate Analysis (Age)

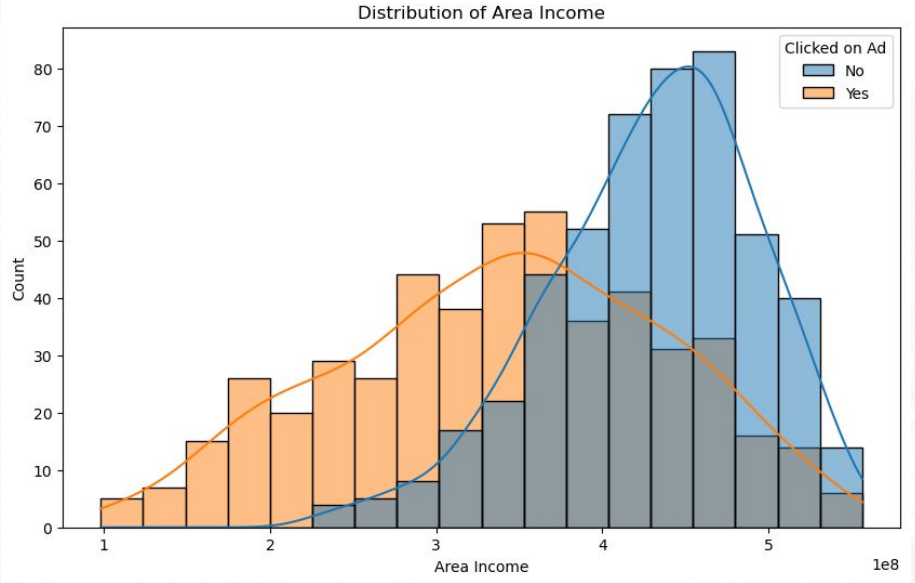
- Younger users (20-40) are less likely to click on ads, with a peak of non-clickers around 30 years old.
- Older users (40-60) are more likely to click on ads, with the highest proportion of clickers around 45-50 years old.
- This age distribution implies that ad strategies might be more effective when targeted towards older demographics.



For more information, see jupyter notebook here

EDA: Univariate Analysis (Area Income)

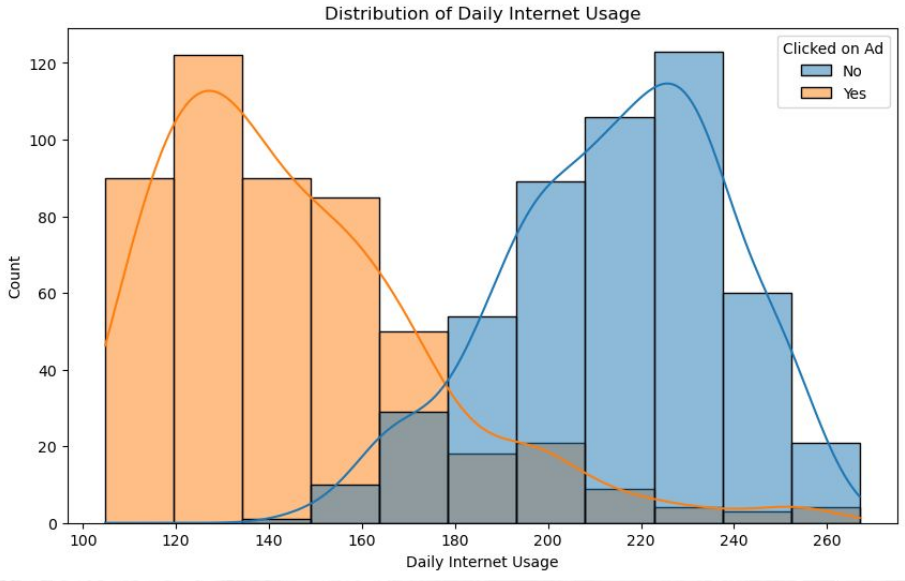
- Non-clickers (blue) are more prevalent in higher income areas, with a peak around 400-500 million.
- Ad-clickers (orange) are more common in lower to middle income areas, peaking around 200-300 million.
- This suggests that ads are more effective in lower to middle income areas, while higher income users are less likely to engage with ads.



For more information, see jupyter notebook here

EDA: Univariate Analysis (Daily Internet Usage)

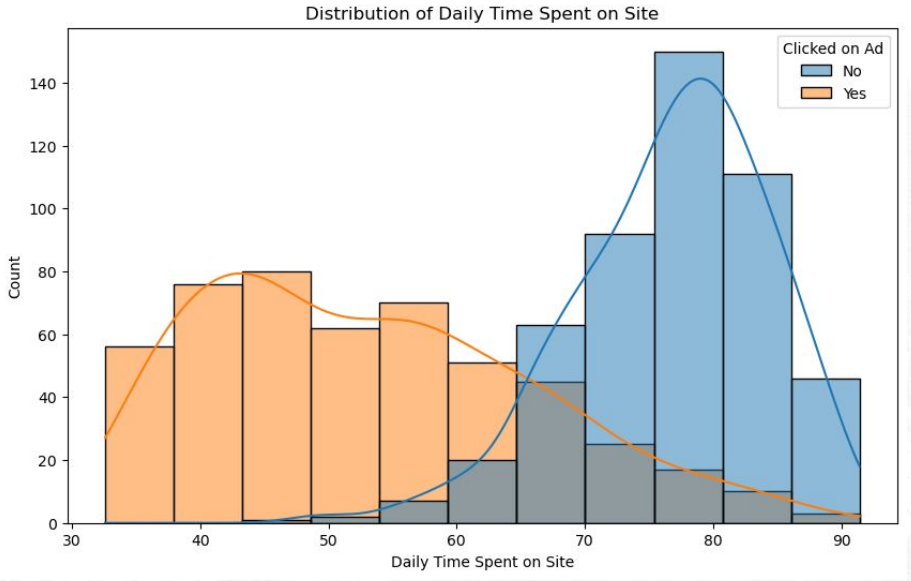
- Non-clickers (blue) have higher daily internet usage, mostly between 200-260 minutes.
- Ad-clickers (orange) show lower daily internet usage, concentrated between 100-160 minutes.
- This indicates that heavy internet users are less likely to click on ads, possibly due to ad fatigue or better ad recognition.



For more information, see jupyter notebook here

EDA: Univariate Analysis (Daily Time Spent on Site)

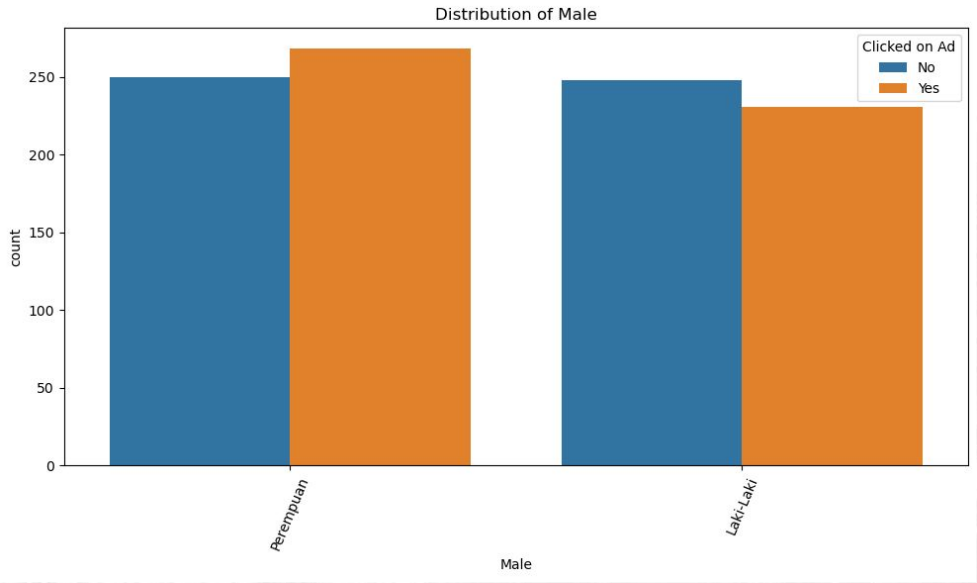
- Users who did not click on ads (blue) tend to spend more time on the site, with the majority spending between 70-90 minutes daily.
- Users who clicked on ads (orange) generally spend less time, with most spending between 30-60 minutes daily.
- There's a clear separation in behavior, suggesting that users who spend more time on the site are less likely to click on ads.



For more information, see jupyter notebook here

EDA: Univariate Analysis (Gender)

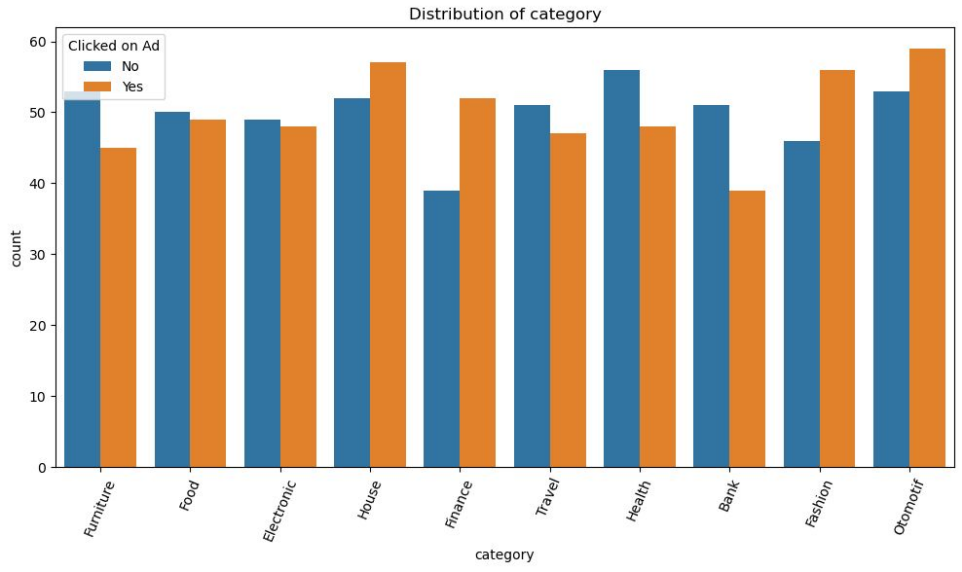
- There are two categories: "Perempuan" ("Female") and "Laki-laki" ("Male").
- "Perempuan" shows slightly higher overall numbers for both clicks and non-clicks.
- The difference between clicks and non-clicks is more pronounced in "Laki-laki".



For more information, see jupyter notebook here

EDA: Univariate Analysis (Category)

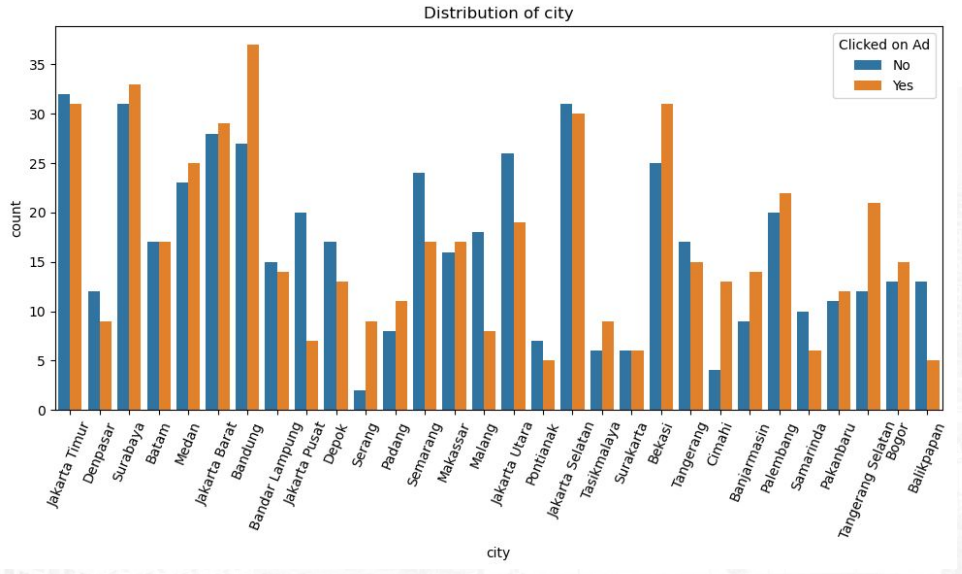
- The "Otomotif" category has the highest number of ad clicks, followed by "Fashion" and "House".
- "Furniture" and "Bank" categories have the lowest ad click rates.
- In most categories, the number of non-clicks (blue) is higher than clicks (orange), with notable exceptions in "House", "Fashion", “Otomotif” and "Finance".
- The "Health" and “Bank” category show the largest gap between clicks and non-clicks, with significantly more non-clicks.



For more information, see jupyter notebook here

EDA: Univariate Analysis (City)

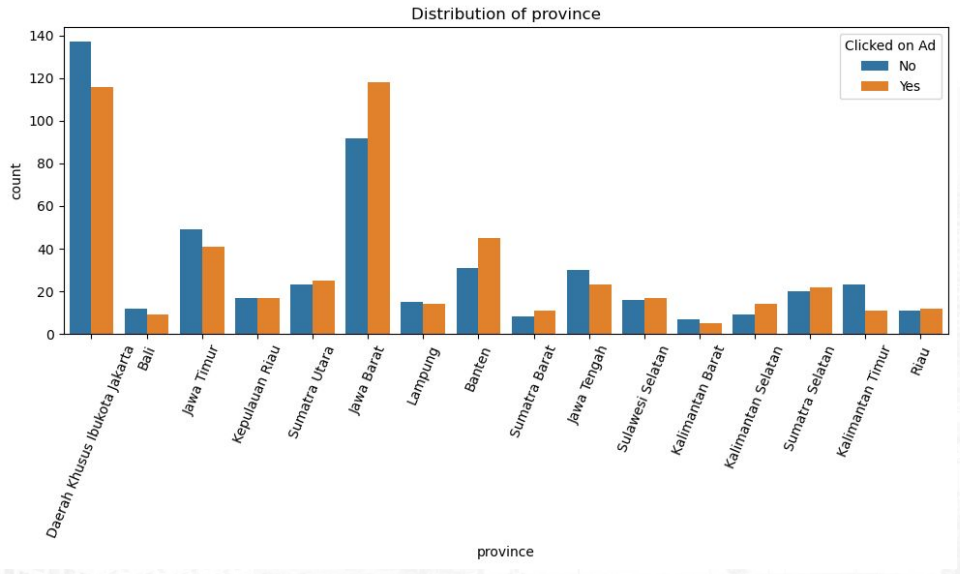
- Bandung has the highest number of clicks while Jakarta Timur with non-clicks.
- Cities like Jakarta Pusat, Jakarta Selatan, and Jakarta Timur also show high engagement.
- Some smaller cities or regions (e.g., Serang, Pekanbaru) have much lower overall counts.
- Click behavior varies significantly between cities, with some showing higher click rates relative to their population.



For more information, see jupyter notebook here

EDA: Univariate Analysis (City)

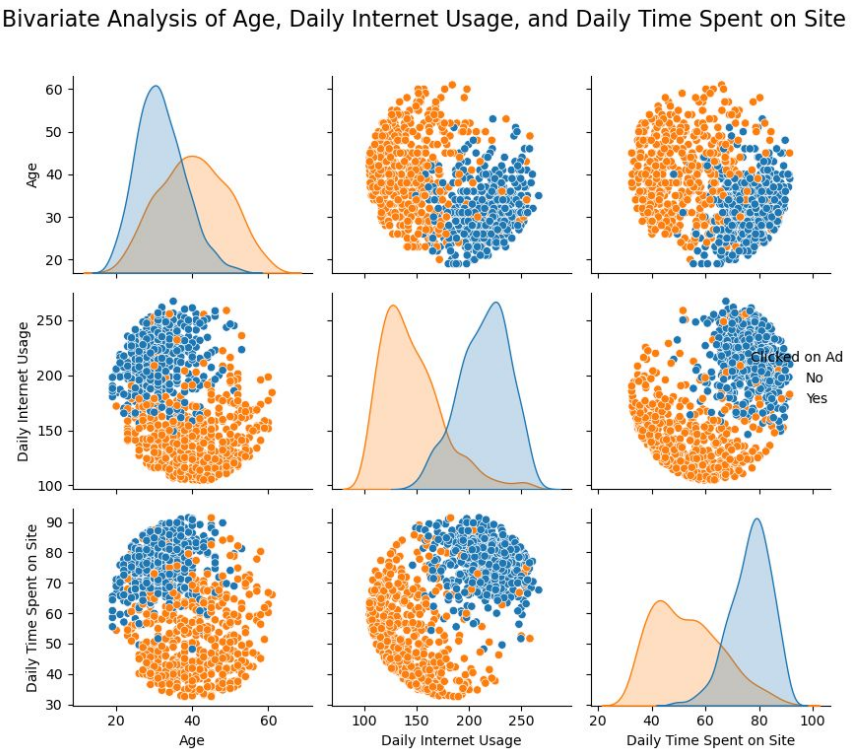
- DKI Jakarta has by far the highest number of both clicks and non-clicks.
- Jawa Barat and Banten also show significant engagement, but much less than DKI Jakarta.
- Many provinces have relatively low engagement numbers.
- The ratio of clicks to non-clicks varies between provinces, potentially indicating regional differences in ad effectiveness.



For more information, see jupyter notebook here

EDA: Bivariate Analysis (Age, Daily Internet Usage, Daily Time Spent on Site)

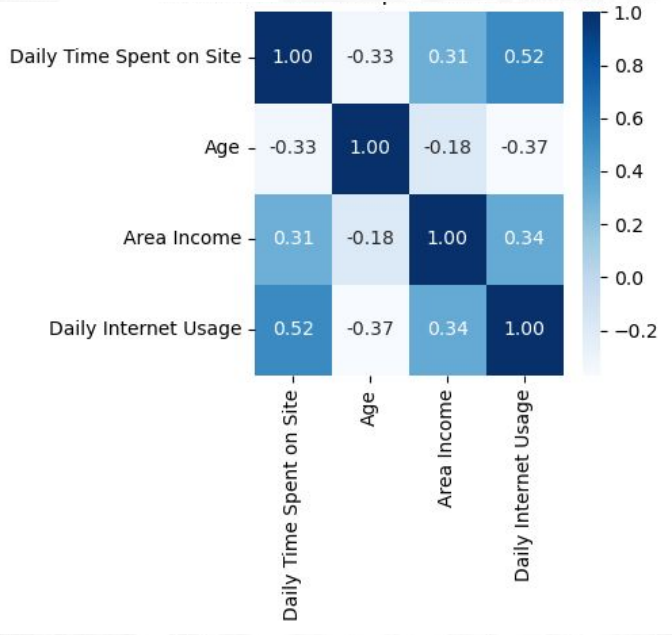
- Age vs. Daily Internet Usage: A weak positive correlation is observed between age and daily internet usage. As age increases, there's a slight tendency for daily internet usage to increase as well.
- Age vs. Daily Time Spent on Site: A very weak negative correlation is observed between age and daily time spent on site. Older individuals tend to spend slightly less time on the site compared to younger individuals.
- Daily Internet Usage vs. Daily Time Spent on Site: A moderate positive correlation is observed between daily internet usage and daily time spent on site. Individuals who use the internet more frequently tend to spend more time on the site.



For more information, see jupyter notebook here

EDA: Bivariate Analysis (Age, Daily Internet Usage, Daily Time Spent on Site)

- There is a strong positive correlation (0.52) between Daily Time Spent on Site and Daily Internet Usage, indicating that individuals who spend more time on the site tend to also use the internet more frequently.
- There is a weak negative correlation (-0.33) between Daily Time Spent on Site and Age, suggesting that older individuals tend to spend slightly less time on the site compared to younger individuals.
- There is a moderate negative correlation (-0.37) between Daily Internet Usage and Age, indicating that older individuals tend to use the internet slightly less frequently compared to younger individuals.
- Overall, these findings suggest that individuals who use the internet more frequently are also more likely to spend more time on the site, regardless of their age or area income. However, age and area income may have some minor influences on internet usage and time spent on site.



For more information, see jupyter notebook here

Data Cleaning & Preprocessing

- In terms of cleaning data, checking on ***missing values*** and ***duplicate*** rows are done
  - No duplicates detected after running the code below.
  - For missing data:
    - Daily Time Spent on Site - fill with mean
    - Daily Internet Usage - fill with mean
    - Area Income - fill with median
    - Male - fill with 'Laki-Laki' as to match the number with 'Perempuan'
  - Column 'Male' is renamed 'Sex' to avoid confusion.

```
duplicate_row = df[df.duplicated(keep=False)]
nan_rows = df[df.isna().any(axis=1)]
df['Daily Time Spent on Site'] = df['Daily Time Spent on Site'].fillna(df['Daily Time Spent on Site'].mean())
df['Daily Internet Usage'] = df['Daily Internet Usage'].fillna(df['Daily Internet Usage'].mean())
df['Area Income'] = df['Area Income'].fillna(df['Area Income'].median())
df['Male'] = df['Male'].fillna('Laki-Laki')
df = df.rename(columns={'Male': 'Sex'})
```

For more information, see jupyter notebook here

- Next for data preprocessing is to handle outliers using the code below:

```
def remove_outliers_iqr(df, columns):
    for col in columns:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]
    return df

df = remove_outliers_iqr(df, nums)
```

For more information, see jupyter notebook here

Data Cleaning & Preprocessing

- In terms of **extracting datetime data**, new features to get, year, month, week, and, day are engineered as below:

```
df['Timestamp'] = pd.to_datetime(df['Timestamp'])


df['year'] = df['Timestamp'].dt.year
df['month'] = df['Timestamp'].dt.month
df['week'] = df['Timestamp'].dt.isocalendar().week
df['day'] = df['Timestamp'].dt.day
```

- df[‘Timestamp’] then dropped for machine learning purposes as it is an object type of feature.

```
df = df.drop('Timestamp', axis=1)
```

For more information, see jupyter notebook here

Data Cleaning & Preprocessing

- Next for data preprocessing, ***feature encoding*** and ***feature standardisation*** are done.
  - Feature encoding for categorical features include:
    - Label encoding for ordinal data (Sex, Clicked on Ad)
    - One hot encoding for non-ordinal data (city, province, category)
  - Feature standardisation using StandardScaler and MinMaxScaler, features has been checked for standardised
    - 'Daily Time Spent on Site' and 'Daily Internet Usage' roughly normal but slightly skewed, standard scaling (Z-score normalization) is used.
    - 'Area Income' is rightly skewed, log transformation followed by standard scaling, which can help reduce the skewness before standardization.
    - Age' to be scaled with Min-Max scaling as it will preserve the relative differences while scaling all values to a 0-1 range.

For more information, see jupyter notebook here

Data Cleaning & Preprocessing

- Next for data preprocessing is to Split Data for train and test using the code below:

```
from sklearn.model_selection import train_test_split


X = df.drop('Clicked on Ad', axis=1)
y = df['Clicked on Ad']
# 'Clicked on Ad' value count
print("Distribution of target 'Clicked on Ad'")
print(y.value_counts())


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

For more information, see jupyter notebook here

Data Modeling

- For this particular project, the company wants to understand how well the advertisement reaches its target audience, so they can attract customers to click on the ad. While focusing on false positive indicate that avoid showing ads to uninterested customers, this project will focused on **lower false negative number** to ensure capturing every possible click.
- Metrics that is used to examine the model are as below:
  - **Recall**: High recall means the model is good at finding most of the users who would click on ads.
  - **F1-Score**: A high F1-score in means the model is doing well at both identifying likely clickers and not missing too many potential clickers.
  - **ROC AUC**: ability of model to differentiate between clickers and non-clickers.
  - **Cross Validation**: evaluate the robustness of model by dividing data into subsets.

For more information, see jupyter notebook here

Rakamin Academy

Modeling Result for Experiment 1 (Before Normalisation/Standardisation)

Model	Recall	F1-Score	ROC AUC	Cross Validation
Logistic Regression	.94 (.94)	.92 (.93)	.98 (.97)	94 (.94)
Decision Tree	.95 (1.00)	.95 (1.00)	.94 (1.00)	.93 (1.00)
Random Forest	.96 (1.00)	.97 (1.00)	.99 (1.00)	.97 (1.00)

For more information, see jupyter notebook here

Rakamin Academy

Modeling Result for Experiment 1 (Before Normalisation/Standardisation)

Model	Recall	F1-Score	ROC AUC	Cross Validation
KNN	.74 (.81)	.67 (.78)	.63 (.86)	.70 (.81)
XGBoost	*.97 (1.00)*	*.97 (1.00)*	*.99 (1.00)*	*.96 (1.00)*

For more information, see jupyter notebook here

Rakamin Academy

Modeling Result for Experiment 2 (After Normalisation/Standardisation)

Model	Recall	F1-Score	ROC AUC	Cross Validation
Logistic Regression	.98 (.98)	.97 (.97)	.99 (.99)	94 (.94)
Decision Tree	.96 (1.00)	.96 (1.00)	.95 (1.00)	.92 (1.00)
Random Forest	.96 (1.00)	.97 (1.00)	.99 (1.00)	.97 (1.00)

For more information, see jupyter notebook here

Rakamin Academy

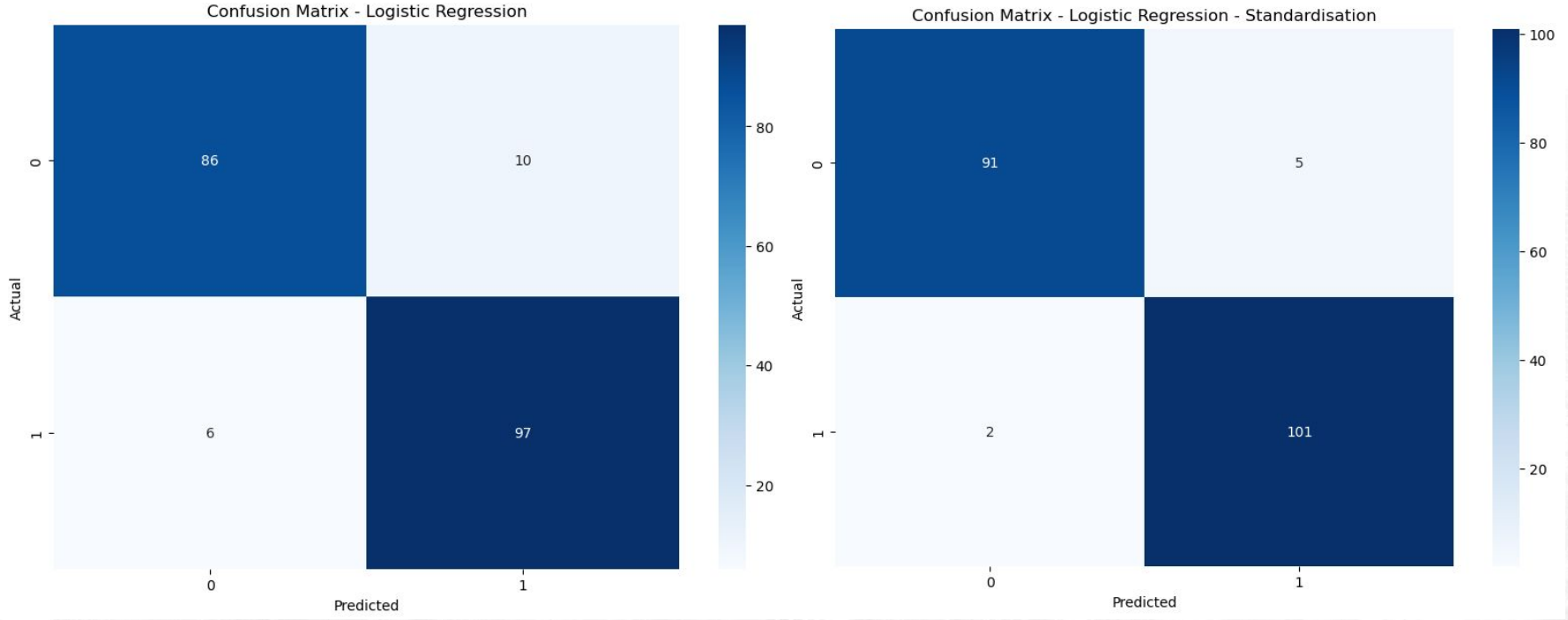
Modeling Result for Experiment 2 (After Normalisation/Standardisation)

Model	Recall	F1-Score	ROC AUC	Cross Validation
KNN	.97 (.98)	.91 (.94)	.96 (.99)	.70 (.81)
XGBoost	.96 (1.00)	.96 (1.00)	.99 (1.00)	.96 (1.00)

For more information, see jupyter notebook here

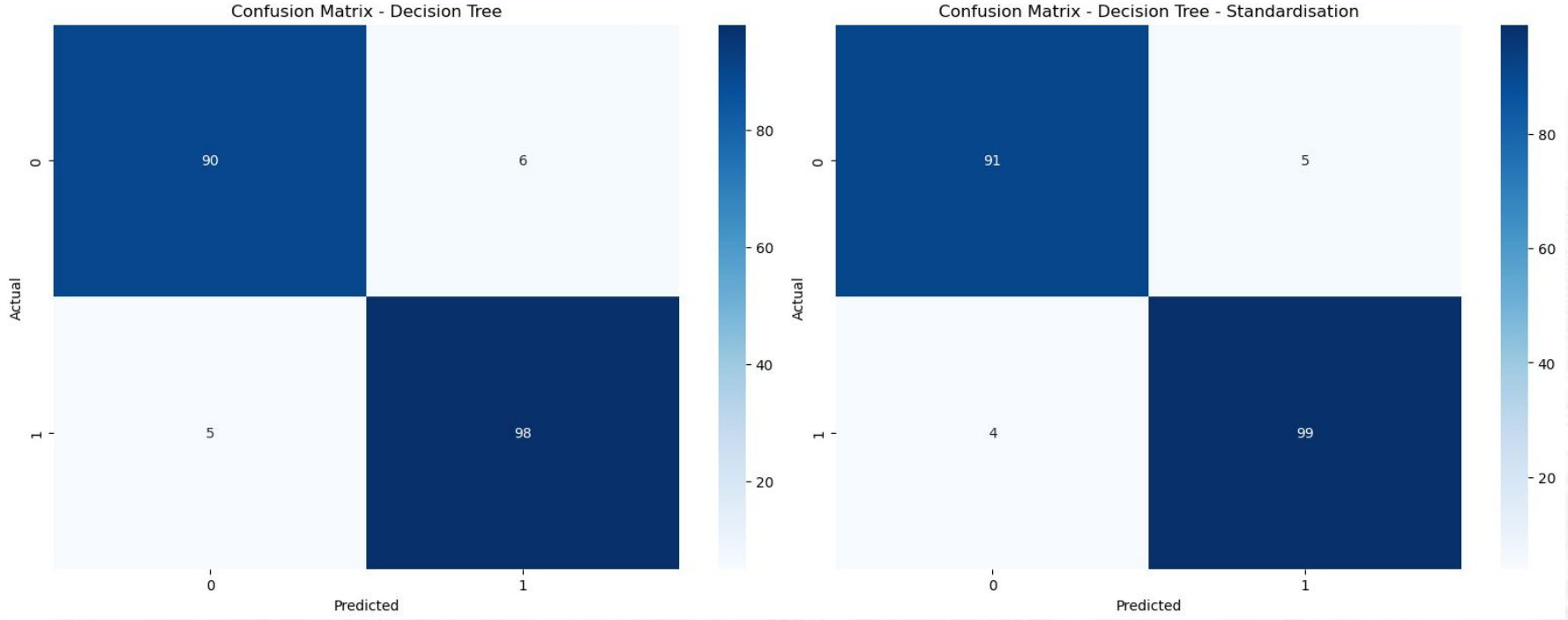
Rakamin Academy

Confusion Matrix: Logistic Regression



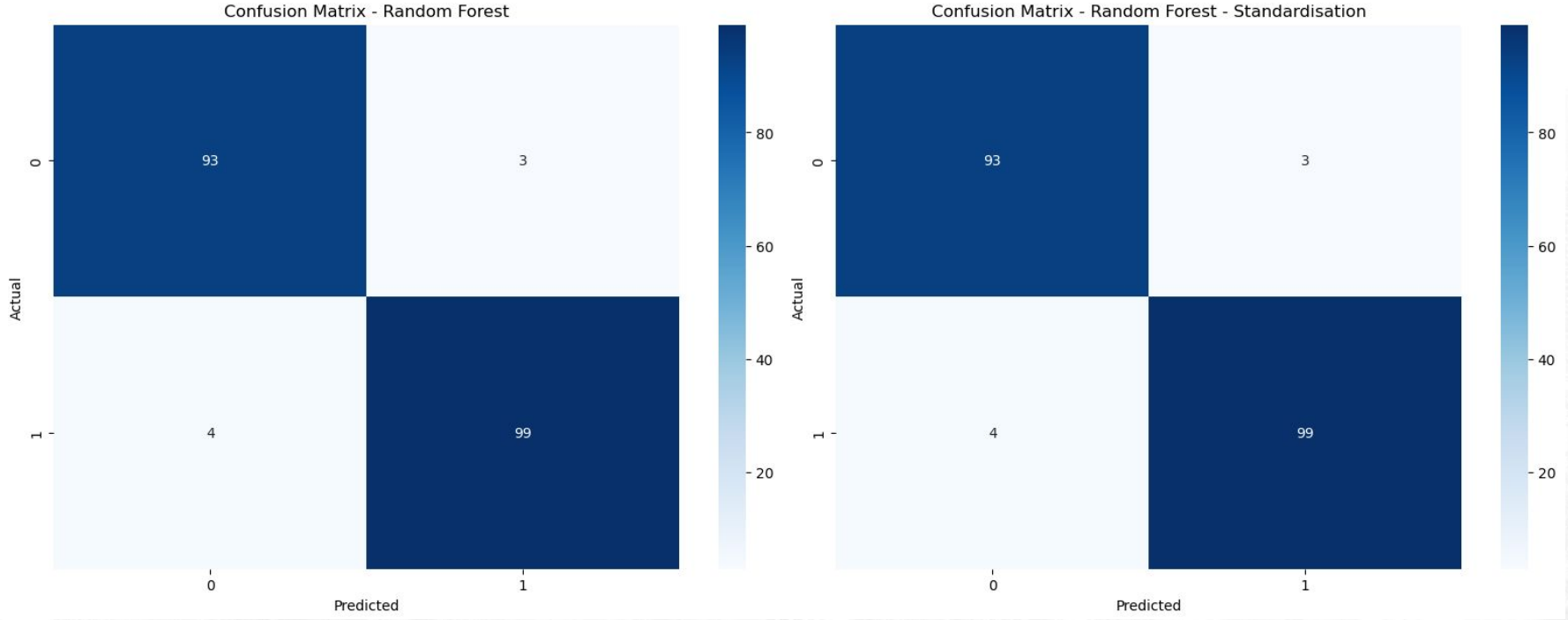
For more information, see jupyter notebook here

Confusion Matrix: Decision Tree



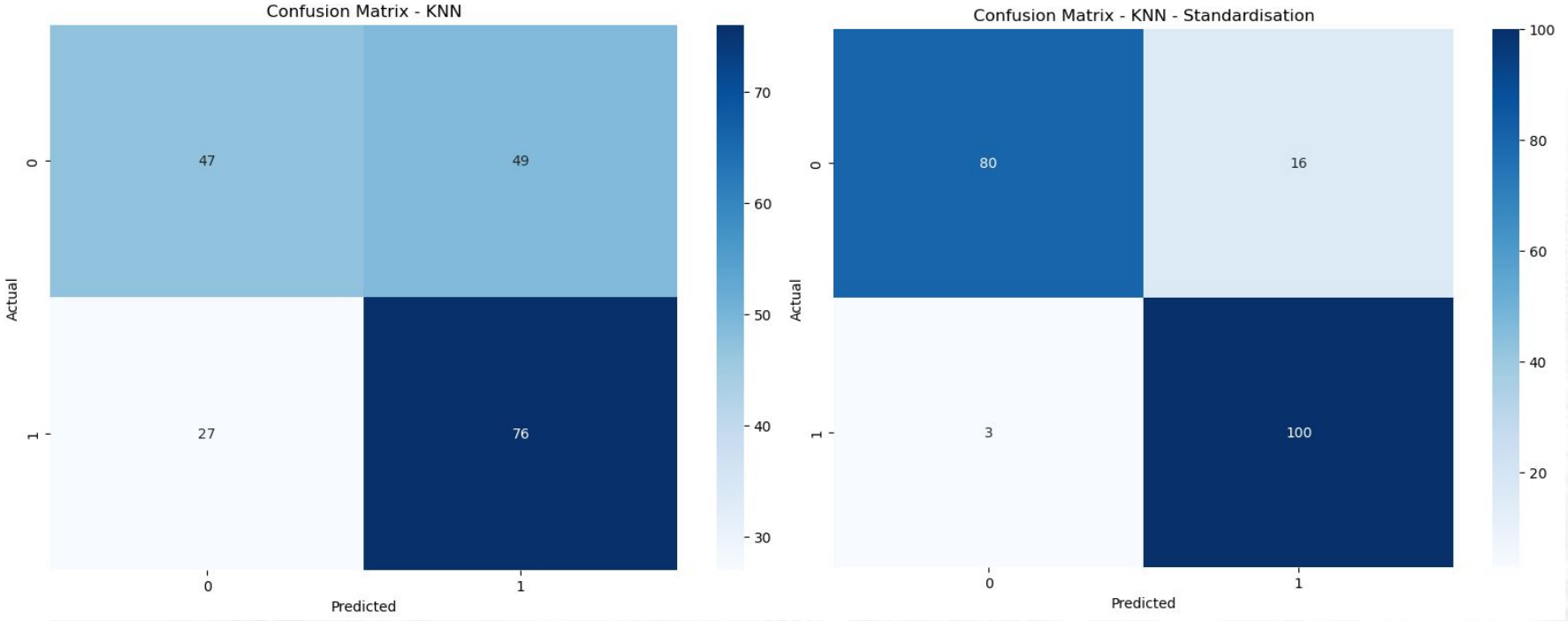
For more information, see jupyter notebook here

Confusion Matrix: Random Forest



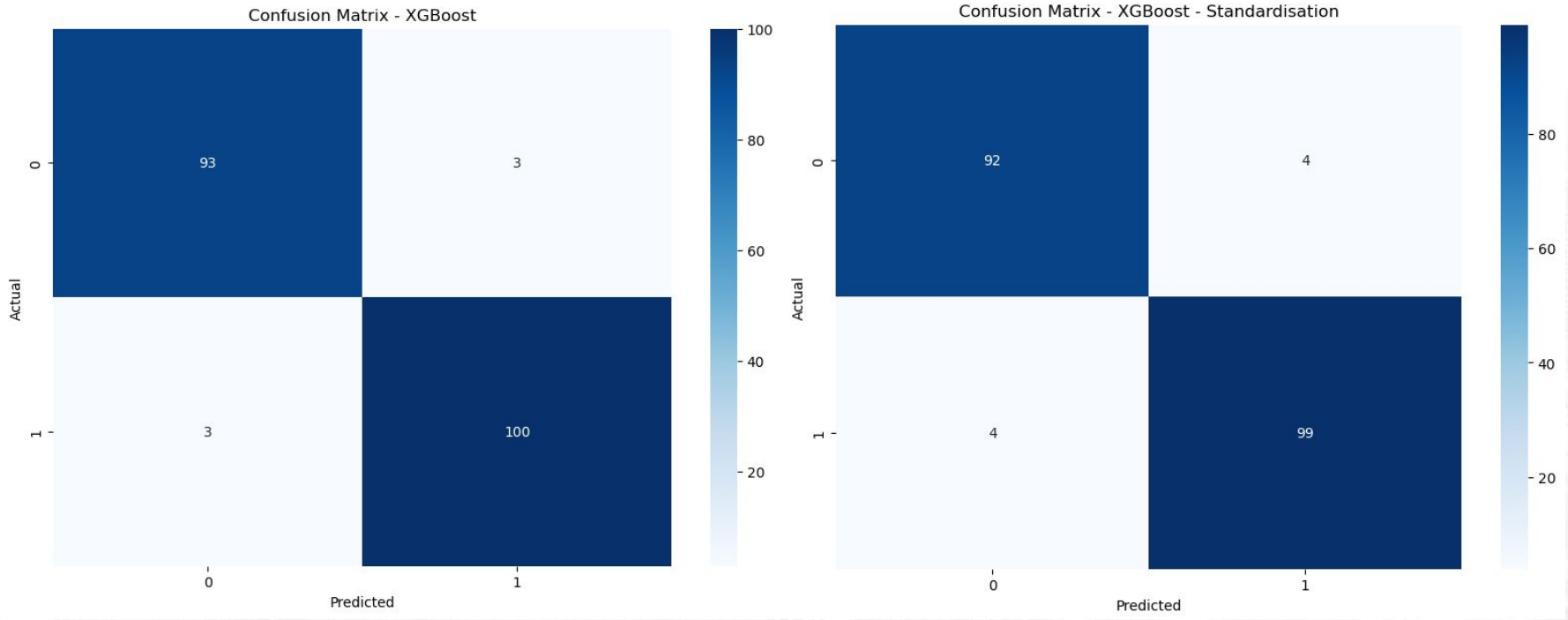
For more information, see jupyter notebook here

Confusion Matrix: KNN



For more information, see jupyter notebook here

Confusion Matrix: XGBoost



For more information, see jupyter notebook here

Best Model Analysis

1. Before Standardisation:
    a. XGBoost and Random Forest show the best overall performance, with high recall, F1-score, ROC AUC, and cross-validation scores.
    b. XGBoost slightly edges out Random Forest with a recall of 0.97 vs 0.96.
    c. KNN performs poorly compared to other models, especially in recall (0.74).
2. After Standardisation:
    a. Logistic Regression shows significant improvement, achieving the highest recall (0.98) among all models.
    b. KNN improves dramatically, with recall increasing from 0.74 to 0.97.
    c. XGBoost, Random Forest, and Decision Tree maintain high performance but don't show significant improvements.
3. False Negative Analysis:
    a. Random Forest has the lowest and most consistent FN count (3 before and after standardization).
    b. XGBoost is close behind with 3 FN before and 4 after standardization.
    c. Logistic Regression improves from 10 to 5 FN after standardization.
    d. KNN shows the most dramatic improvement, from 49 to 16 FN, but still has the highest FN count.
    e. Decision Tree is relatively consistent with 6 and 5 FN.
4. Most Balanced Model:
    a. XGBoost and Random Forest consistently perform well across all metrics and have low FN counts in both scenarios.
    b. Logistic Regression becomes highly competitive after standardization, with high recall and significant reduction in FN.

For more information, see jupyter notebook here

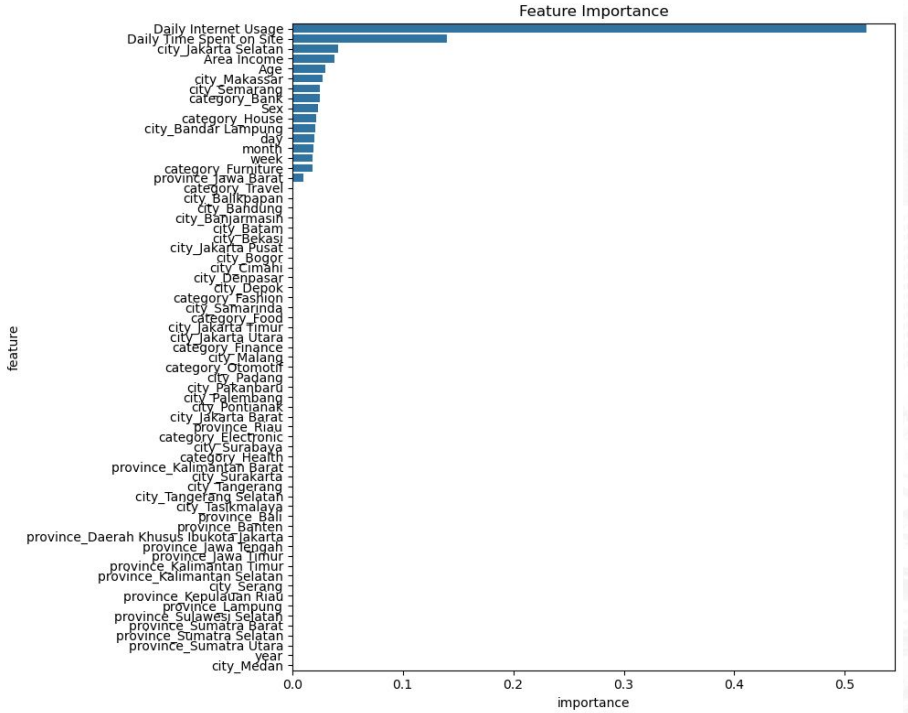
Conclusion

XGBoost appears to be the best model both before and after standardisation due to its consistently high performance across all metrics (recall, F1-score, ROC AUC, and cross-validation). XGBoost is a solid choice for a model that performs well regardless of standardisation. However, if simplicity and interpretability are important, Logistic Regression shows significant improvement after standardisation and could also be a good option. Since extreme changes in metrics observed in Logistic Regression which might indicate overfitting, XGBoost is selected to be the best model afterall.

For more information, see jupyter notebook here

Feature Importance

From the graph, it can be seen that '**Daily Internet Usage**' and '**Daily Time Spent on Site**' are the top two features that influence customers to click on the ads. While both also correlated, both features are retained as they are not redundant (r=.52) and different features based on domain knowledge. Hence, we can accept the two features influencing the target.



For more information, see jupyter notebook here

Key Insights from EDA and Feature Importance

- User behavior, particularly internet usage and time spent on site, strongly influences ad engagement.
- Geographic location plays a significant role in predicting ad interactions.
- Economic factors, such as area income, affect how users respond to advertisements.
- Some product or service categories are more effective at driving ad clicks than others.
- Demographic factors have less impact on ad engagement than initially thought.
- The timing of ad delivery, including day and month, can influence user response.
- Regional differences in ad engagement suggest the need for localized strategies.
- Broad targeting approaches may be more effective than highly specific ones for many features.
- There's potential for effective user segmentation based on behavior, location, and economic factors.
- Ongoing data analysis is crucial to refine and adapt marketing strategies over time.

For more information, see jupyter notebook here

Business Recommendation & Simulation

Insights	Recommendations	Example
Optimize for Internet Usage and Site Engagement	Focus on strategies that target users based on their internet usage patterns and time spent on site.	- Develop personalized ad experiences based on users' daily internet usage - Create content and features that encourage longer site visits - Implement a tiered engagement program rewarding users for time spent on site
Geotargeting with focus on Jakarta Selatan	Prioritize ad campaigns and user acquisition efforts in Jakarta Selatan while maintaining presence in other key cities.	- Create location-specific campaigns for Jakarta Selatan - Analyze user behavior in Jakarta Selatan to identify unique characteristics *(data team)* - Use insights from Jakarta Selatan to improve strategies in other cities
Income-based targeting	Develop strategies that consider users' area income levels.	- Segment audiences based on area income data - Create tailored ad content and offers for different income segments - Test pricing strategies that align with various income levels

For more information, see jupyter notebook here

Business Recommendation & Simulation

Insights	Recommendations	Example
City-specific strategies	Develop targeted approaches for high-importance cities like Makassar and Semarang.	- Conduct in-depth analysis of user behavior in these cities *(data team)* - Create city-specific marketing campaigns and promotions - Establish local partnerships to increase brand presence in these areas
Category optimization	Focus on key categories like Banking and House, while also maintaining a diverse category presence.	- Allocate more resources to creating compelling ads for 'Bank' and 'House' categories - Analyze user journey and conversion paths in these top categories *(data team)* - Develop cross-category promotion strategies to leverage high-performing categories
Demographic considerations	While age doesn't appear as a top feature, consider sex as a relevant factor in ad targeting.	- Develop gender-specific ad creatives and messaging - Analyze and optimize ad performance separately for different genders - Test gender-neutral campaigns to compare effectiveness

For more information, see jupyter notebook here

Business Recommendation & Simulation

Insights	Recommendations	Example
Temporal targeting	Consider day and month in ad scheduling and campaign planning.	- Analyze ad performance patterns by day of the week and month *(data team)* - Develop day-specific or seasonal ad campaigns - Optimize ad delivery times based on when users are most likely to engage
Continuous monitoring and adaptation (*data team)*	Regularly reassess feature importance and adjust strategies accordingly.	- Implement a system for ongoing analysis of feature importance - Establish a process for quickly adapting marketing strategies based on changing feature importance - Conduct A/B tests to validate the impact of strategy changes based on feature importance

For more information, see jupyter notebook here

Simulation

- Assumption for cost and revenue has been made with as such
  - marketing_cost_per_user = 5000 (Assumed Rp 5000 per user)
  - revenue_per_conversion = 50000 (Assumed Rp 50000 per user conversion)

	Before Modeling	After Modeling	Changes
Conversions	500	513	+ **2.59%**
Cost	Rp4.955.000	Rp4.955.000	
Revenue	Rp25.000.000	Rp25.650.000	
Profit	Rp20.045.000	Rp20.695.000	+ **Rp650.000**

For more information, see jupyter notebook here

Summary

- **Behavior-driven targeting is crucial**: The most influential factors in ad engagement are user behaviors, specifically daily internet usage and time spent on the site. This suggests that personalized, behavior-based ad strategies could be highly effective.
- **Geographic and economic factors matter**: Location-based targeting, especially in key cities like Jakarta Selatan, Makassar, and Semarang, combined with income-based segmentation, can significantly improve ad performance.
- **Category-specific optimization is important**: Focusing on high-performing categories like ‘Bank’ and ‘House’ while maintaining a diverse presence can enhance overall ad effectiveness.
- **Simulation shows potential for improvement**: The modeling approach demonstrated a **2.59% increase in conversions**, resulting in an additional profit of Rp650,000. This indicates that data-driven, targeted approaches can lead to tangible financial benefits.

For more information, see jupyter notebook here