# 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

## Data Modeling

# Modeling Result for Experiment 1 (Before Normalisation/Standardisation)

| Model | Recall | F1-Score | ROC AUC | Cross Validation |
|---|---|---|---|---|
| Logistic Regression | 1.00 (1.00) | .62 (.68) | .71 (.79) | 1.00 (1.00) |
| Decision Tree | .90 (1.00) | .90 (1.00) | .91 (1.00) | .95 (1.00) |
| Random Forest | .92 (1.00) | .93 (1.00) | .98 (1.00) | .97 (1.00) |

For more information, see jupyter notebook here

# Data Modeling

## Modeling Result for Experiment 1 (Before Normalisation/Standardisation)

| Model | Recall | F1-Score | ROC AUC | Cross Validation |
|---|---|---|---|---|
| KNN | .66 (.81) | .62 (.79) | .67 (.86) | .73 (.81) |
| *XGBoost* | *.93 (1.00)* | *.94 (1.00)* | *.98 (1.00)* | *.97 (1.00)* |

For more information, see jupyter notebook here

# Modeling Result for Experiment 2 (After Normalisation/Standardisation)

| Model | Recall | F1-Score | ROC AUC | Cross Validation |
|---|---|---|---|---|
| Logistic Regression | .94 (.98) | .93 (.98) | .97 (.99) | 1.00 (1.00) |
| Decision Tree | .90 (1.00) | .90 (1.00) | .91 (1.00) | .94 (1.00) |
| Random Forest | .91 (1.00) | .92 (1.00) | .98 (1.00) | .97 (1.00) |

For more information, see jupyter notebook here

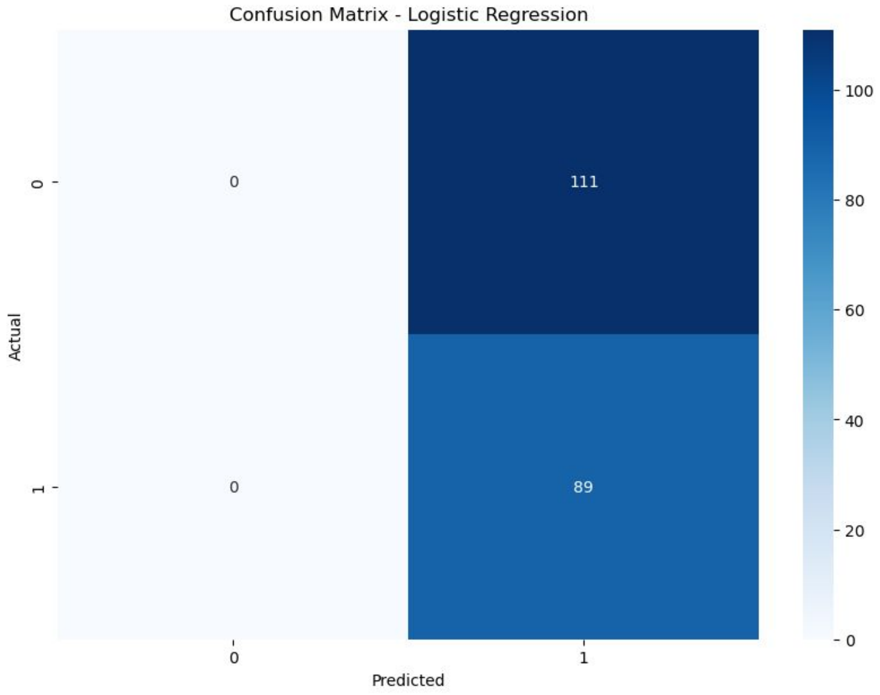
# Data Modeling

## Modeling Result for Experiment 2 (After Normalisation/Standardisation)

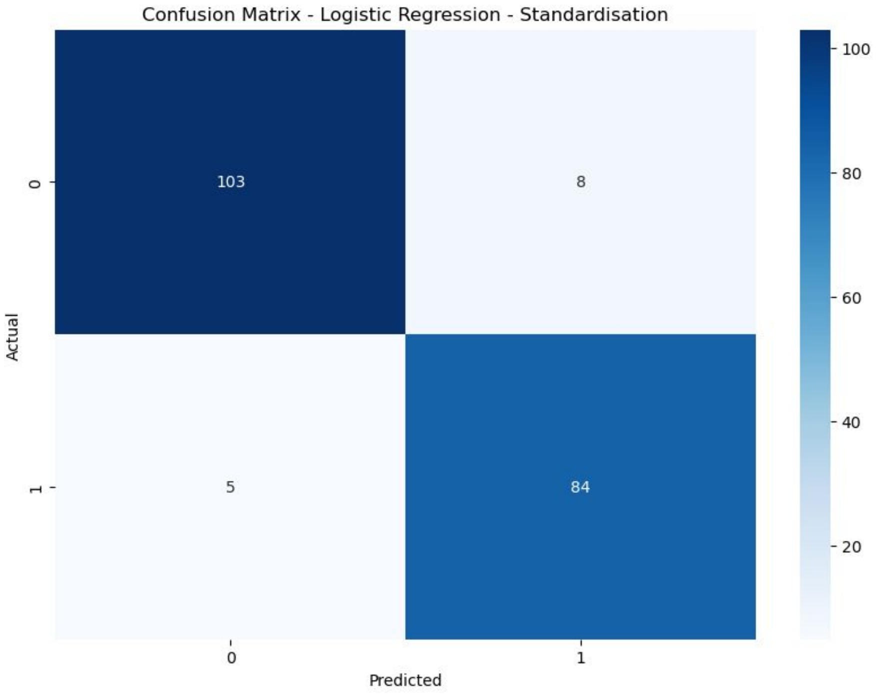
| Model | Recall | F1-Score | ROC AUC | Cross Validation |
|---|---|---|---|---|
| KNN | .96 (1.00) | .89 (.96) | .95 (.99) | .73 (.81) |
| XGBoost | .92 (1.00) | .93 (1.00) | .98 (1.00) | .97 (1.00) |

For more information, see jupyter notebook here

# Confusion Matrix: Logistic Regression



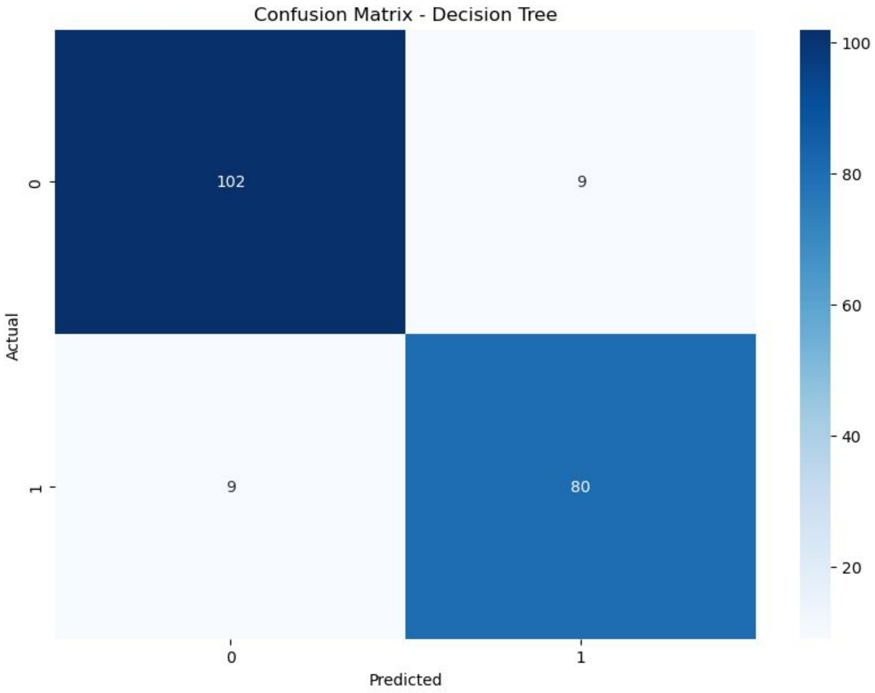
Confusion Matrix - Logistic Regression

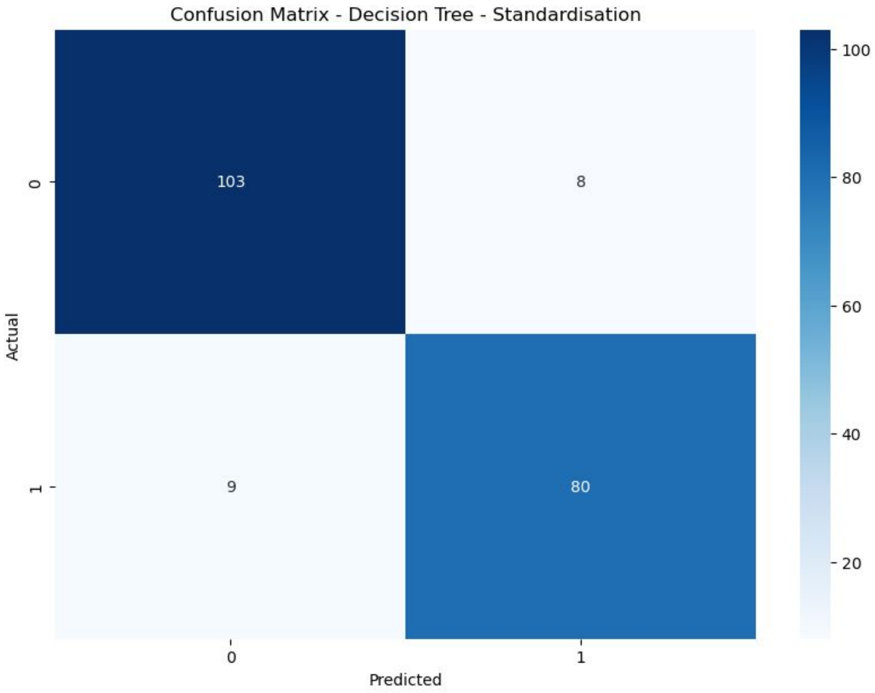


Confusion Matrix - Logistic Regression - Standardisation

For more information, see jupyter notebook here

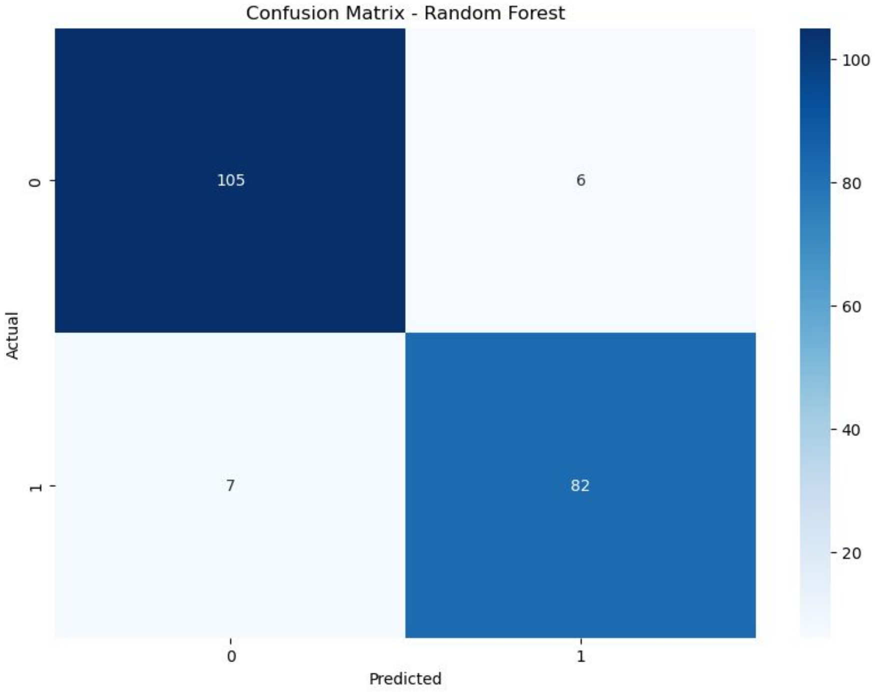
# Confusion Matrix: Decision Tree

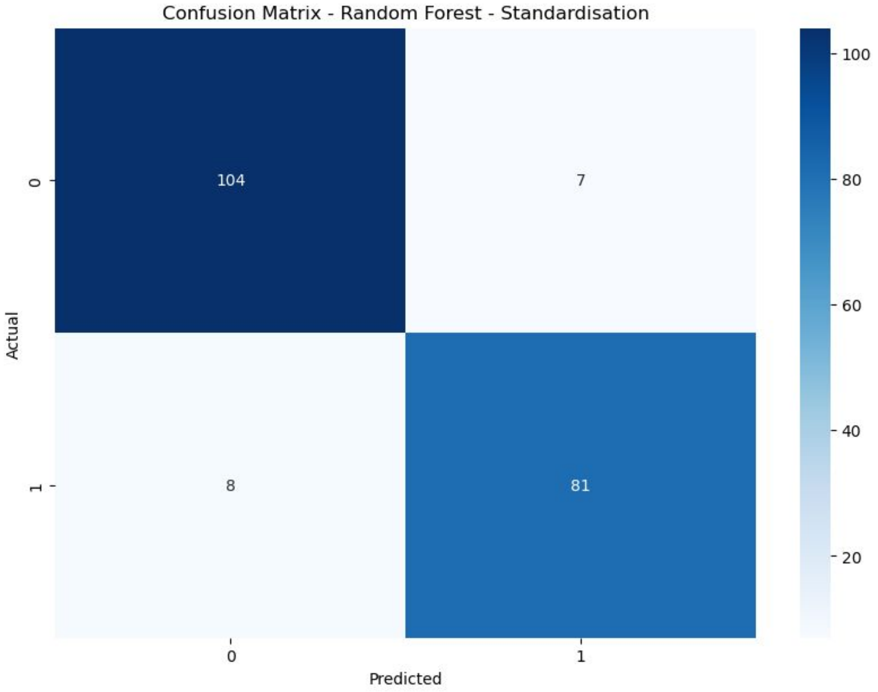




For more information, see jupyter notebook here

# Confusion Matrix: Random Forest

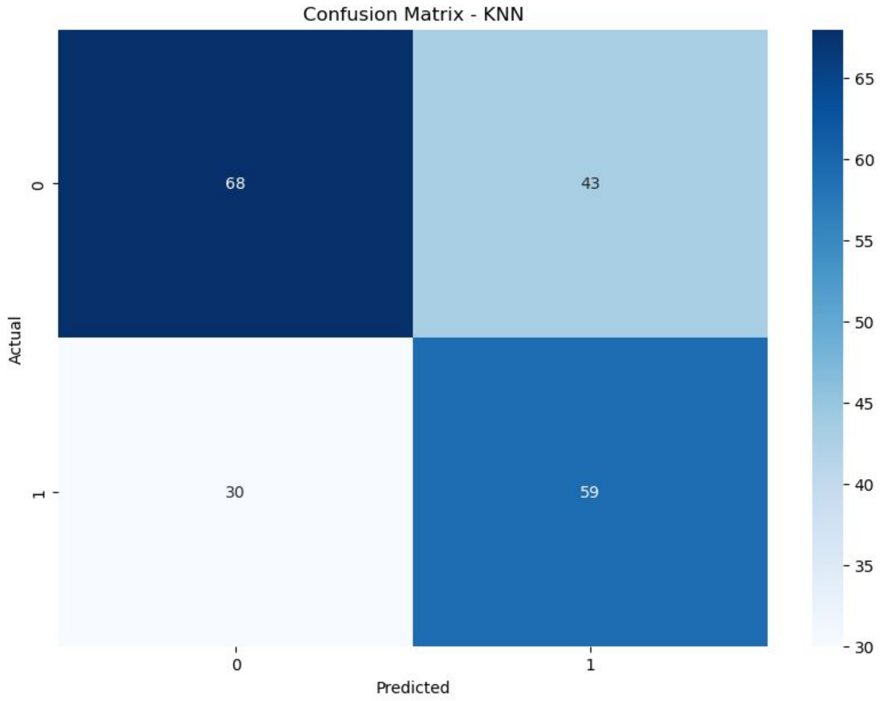


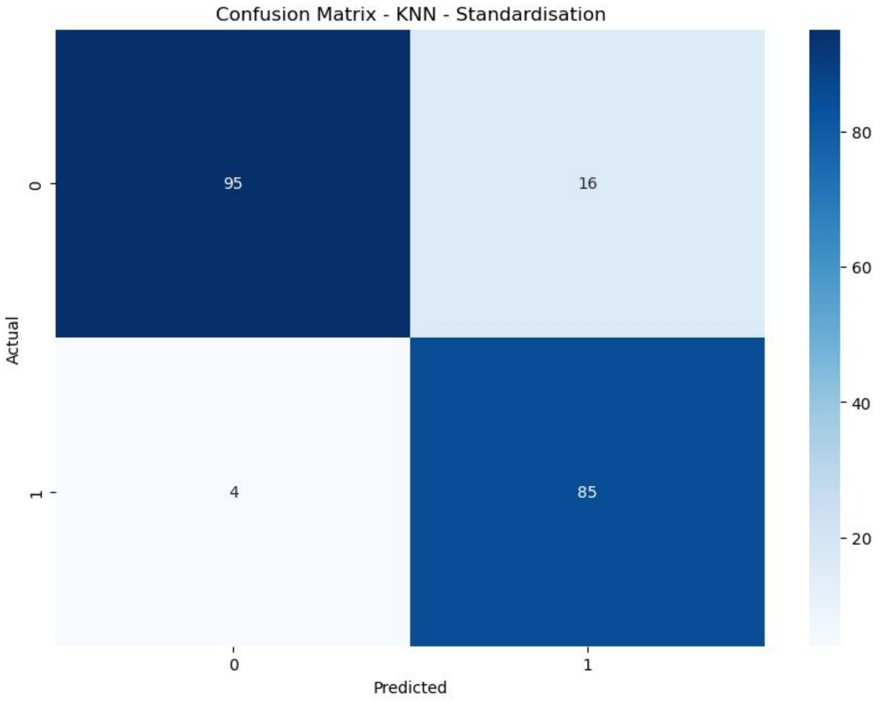


For more information, see jupyter notebook here

# Data Modeling

## Confusion Matrix: KNN

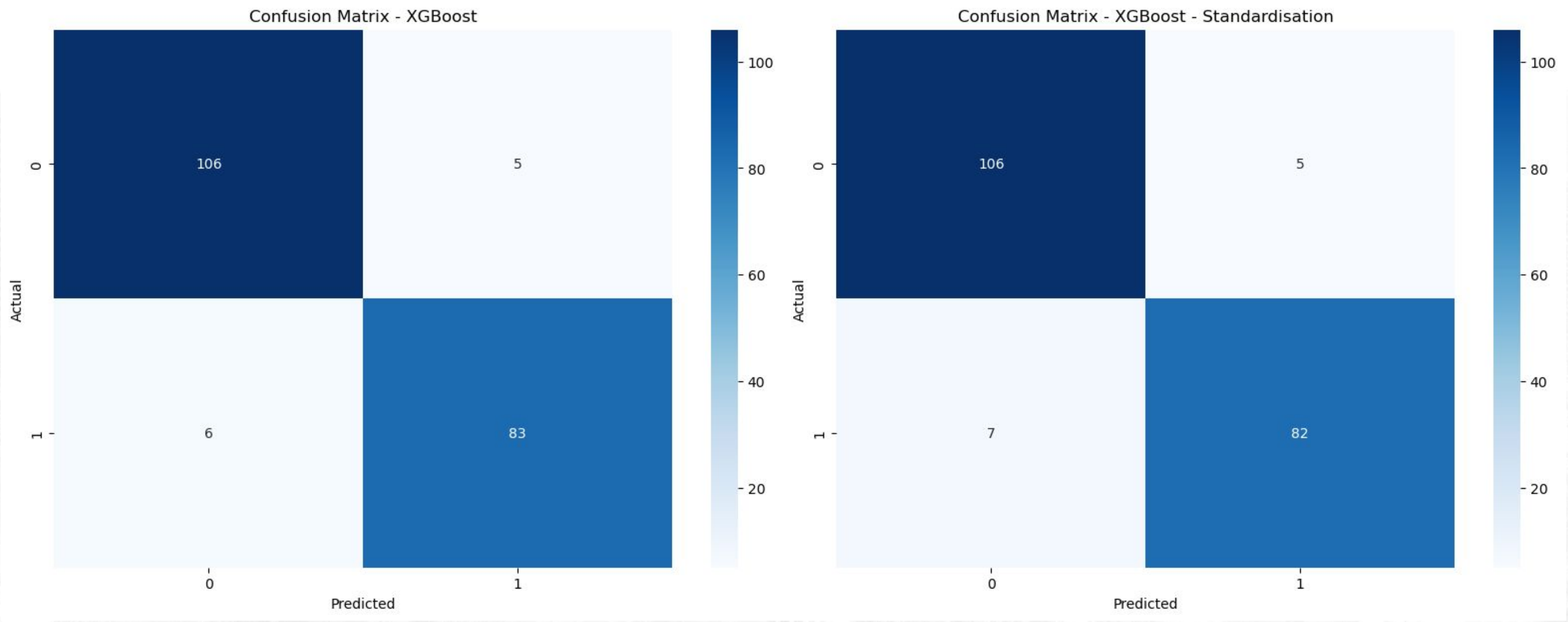




For more information, see jupyter notebook here

# Data Modeling

## Confusion Matrix: XGBoost



Confusion Matrix - XGBoost

Confusion Matrix - XGBoost - Standardisation

For more information, see jupyter notebook here

## Best Model Analysis

1. Before Standardisation:
   a. XGBoost has the best overall performance, with high recall, F1-score, ROC AUC, and cross-validation scores across all metrics.
2. After Standardisation:
   a. XGBoost and Logistic Regression both perform extremely well after standardization.
   b. Logistic Regression improves significantly in terms of F1-score (.93) and ROC AUC (.97).
   c. KNN also improves in terms of recall (from .66 to .96) but has a slightly lower F1-score compared to the other models.
3. Most Balanced Model:
   a. After standardization, XGBoost still performs consistently across all metrics, making it the most reliable model overall.
   b. Random Forest is also a strong competitor in both experiments, but its recall is slightly lower than that of XGBoost after standardization.

Conclusion:

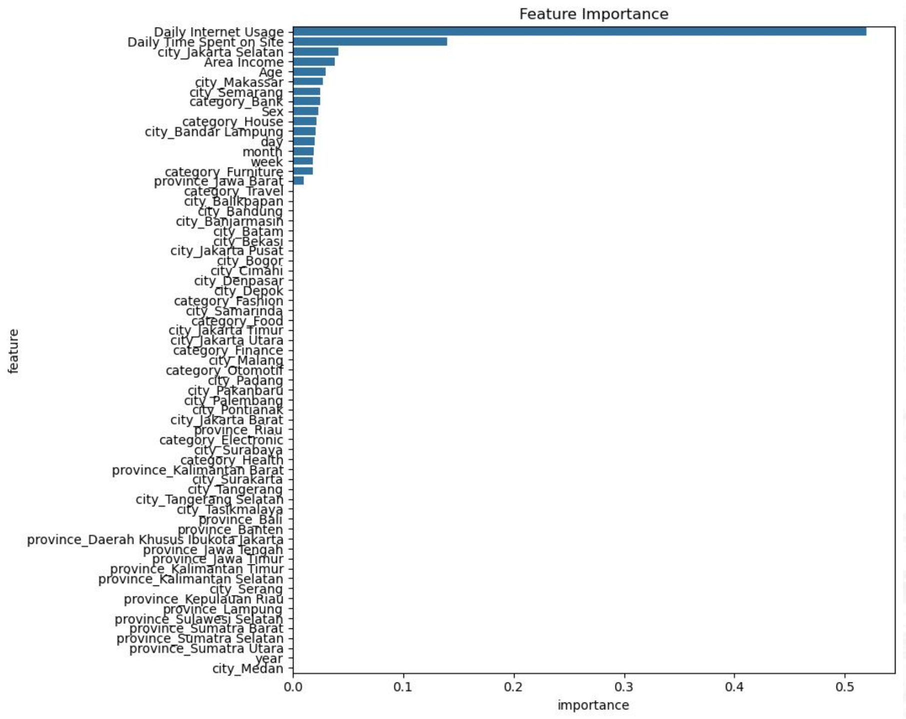
XGBoost appears to be the best model both before and after standardisation due to its consistently high performance across all metrics (recall, F1-score, ROC AUC, and cross-validation). XGBoost is a solid choice for a model that performs well regardless of standardisation. However, if simplicity and 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

# Data Modeling

## 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 customer 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