Predict Clicked Ads Customer Classification by using Machine Learning

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

Rakamin Academy

Created by:
Kenneth Wahyudi, S.Si.

kennethwahyudi48@gmail.com
https://www.linkedin.com/in/kenneth-wahyudi-80b886209/

I am an aspiring data scientist with interest in machine learning and paddling through the data lake. I have experiences as a research intern in BATAN, and several data science and machine learning projects under my belt. Specifically intermediate to advanced knowledge in Exploratory Data Analysis, Data Pre-processing, Supervised & Unsupervised Learning, as well as Data Visualization and Storytelling. And I am always looking forward to exploring new and uncharted territories that might allow myself to grow and improve.

Overview

Overview

In Indonesia, a company aspires to gauge the efficacy of the advertisements they broadcast. This endeavor carries profound significance for the company, allowing them to gauge the extent to which their marketed advertisements resonate. This insight, in turn, serves as a compelling magnet to allure customers into the realm of their advertisements. By meticulously analyzing historical advertisement data and uncovering invaluable insights and recurring patterns, the company can steer its marketing efforts with precision. The focal point of this case is the creation of a machine learning classification model that adeptly identifies the most suitable target customers.

All results are from outputs of codes written in Python and its packages. JupyterLab was the GUI used in this project.

The dataset used in this project can be found in the following link.

Customer Type and Behavior Analysis on Advertisement

In this section, we're focusing on the aspect of exploratory data analysis or EDA. The outlines of the process are as follows :

1. Taking a look at the descriptive statistics
2. Creating a univariate analysis
3. Creating a bivariate analysis
4. Creating a multivariate analysis

Now, let's take a look at the details of the processes outlined above.

The complete code and .ipynb file of the exploratory data analysis can be found in the following link.

Customer Type and Behaviour Analysis on Advertisement

Rakamin Academy

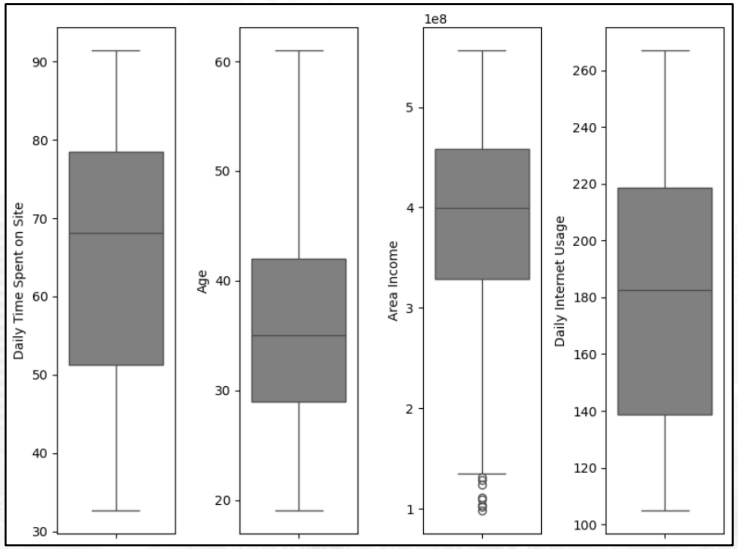
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Unnamed: 0                1000 non-null   int64
 1   Daily Time Spent on Site  987 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
```

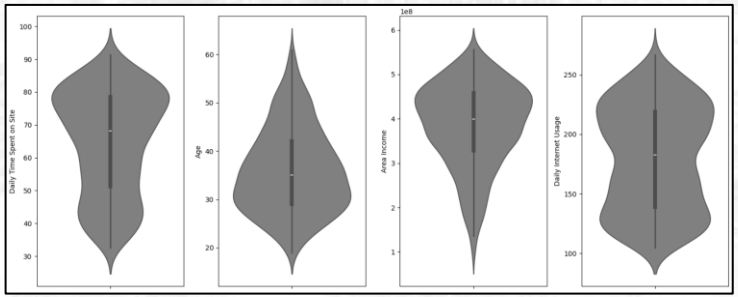
	Unnamed: 0	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage
count	1000.000000	987.000000	1000.000000	9.870000e+02	989.000000
mean	499.500000	64.929524	36.009000	3.848647e+08	179.863620
std	288.819436	15.844699	8.785562	9.407999e+07	43.870142
min	0.000000	32.600000	19.000000	9.797550e+07	104.780000
25%	249.750000	51.270000	29.000000	3.286330e+08	138.710000
50%	499.500000	68.110000	35.000000	3.990683e+08	182.650000
75%	749.250000	78.460000	42.000000	4.583554e+08	218.790000
max	999.000000	91.430000	61.000000	5.563936e+08	267.010000

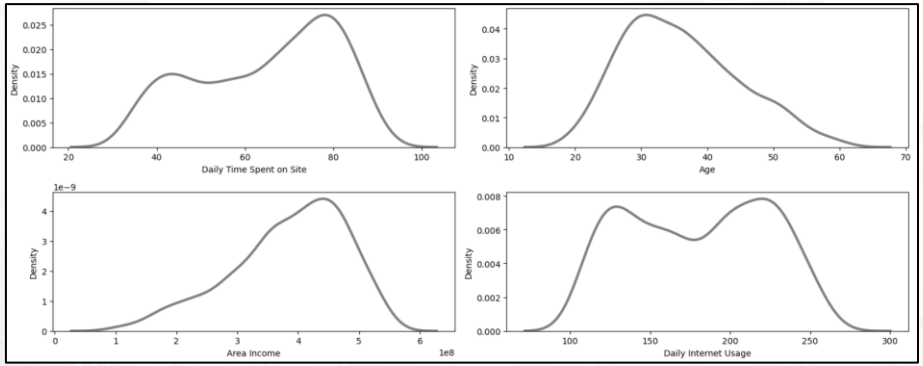
	Male	Timestamp	Clicked on Ad	city	province	category
count	997	1000	1000	1000	1000	1000
unique	2	997	2	30	16	10
top	Perempuan	5/26/2016 15:40	No	Surabaya	Daerah Khusus Ibukota Jakarta	Otomotif
freq	518	2	500	64	253	112

From the info and descriptive statistics to the left, we can see that some columns have NULL values that needs to be addressed later on in the preprocessing stage. We also can see that the target variable 'Clicked on Ad' is perfectly balanced. And the other numerical columns also have a relatively normal distribution, because the mean and median are not too far apart. For the categorical columns, it seems like the values are good since there isn't a massive imbalance in the classes.

Customer Type and Behaviour Analysis on Advertisement

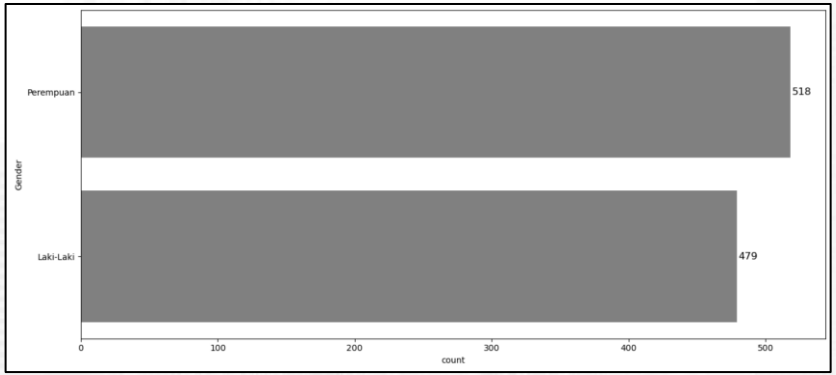


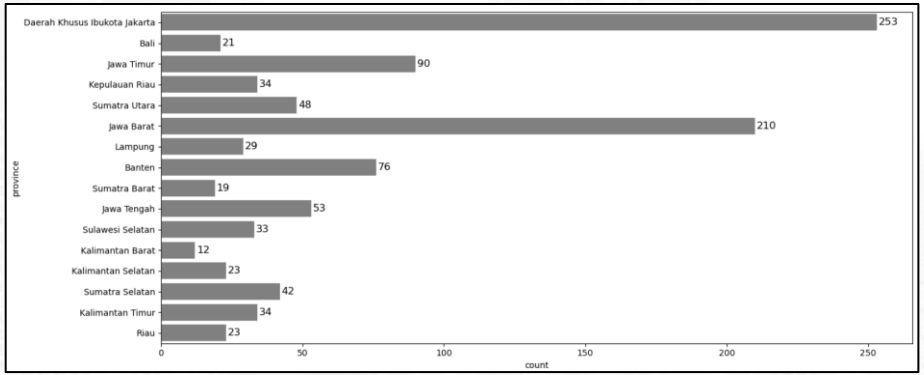


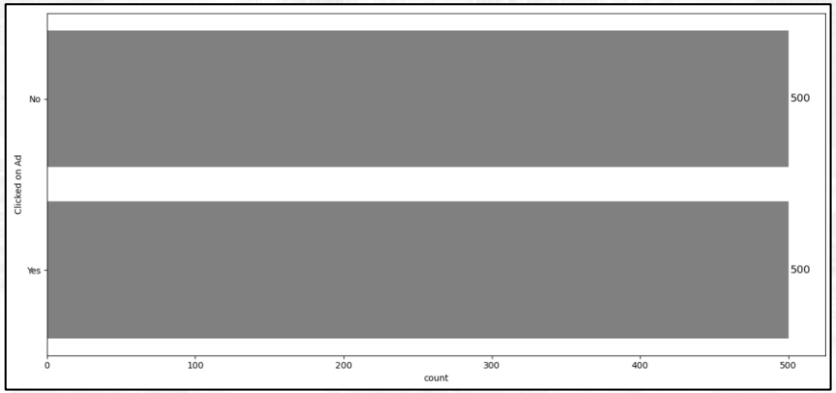


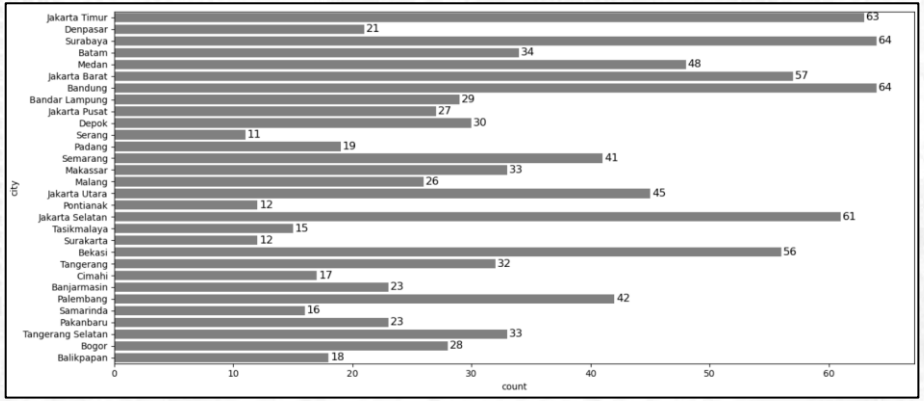
From the boxplot, kdeplot, and violinplot above and to the left, we can see that the 'Age' and 'Area Income' column is slightly skewed. With age being slightly postively skewed, and the area income being slightly negatively skewed. However, the other 2 columns ('Daily Time Spent on Site', and 'Daily Internet Usage') resembles a bimodal distribution, since they have 2 visible peaks. And the area income is the only column that houses a few outliers.

Customer Type and Behaviour Analysis on Advertisement

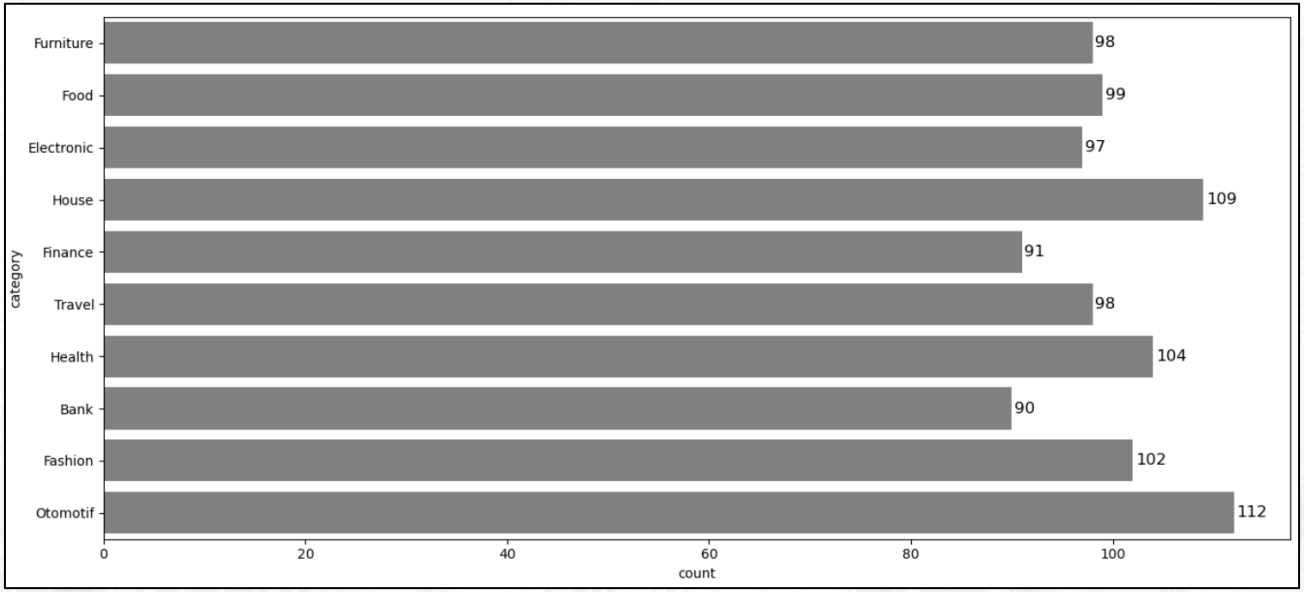






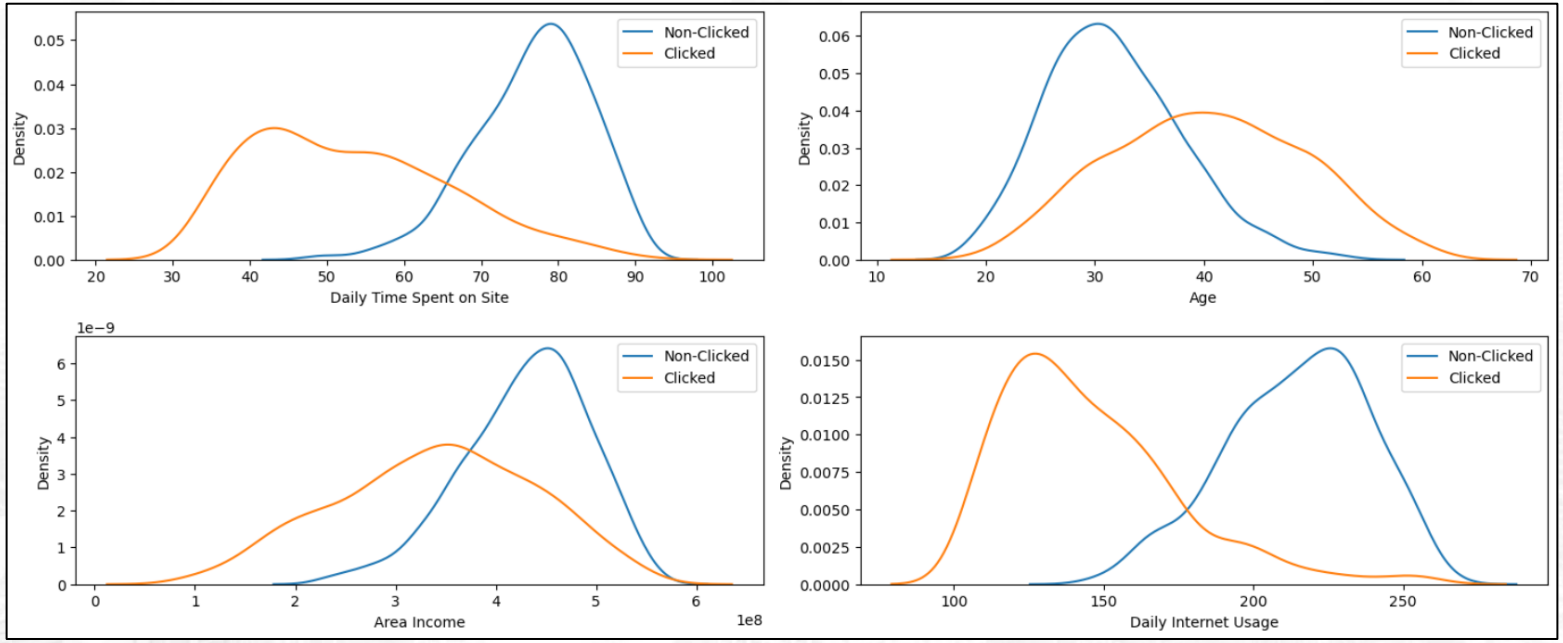


Customer Type and Behaviour Analysis on Advertisement



From the countplot above and in the previous slide, we can see that most columns have little to no imbalance. However, moderate imbalance is present in the 'city' column, and severe imbalance is present in the 'province' column.

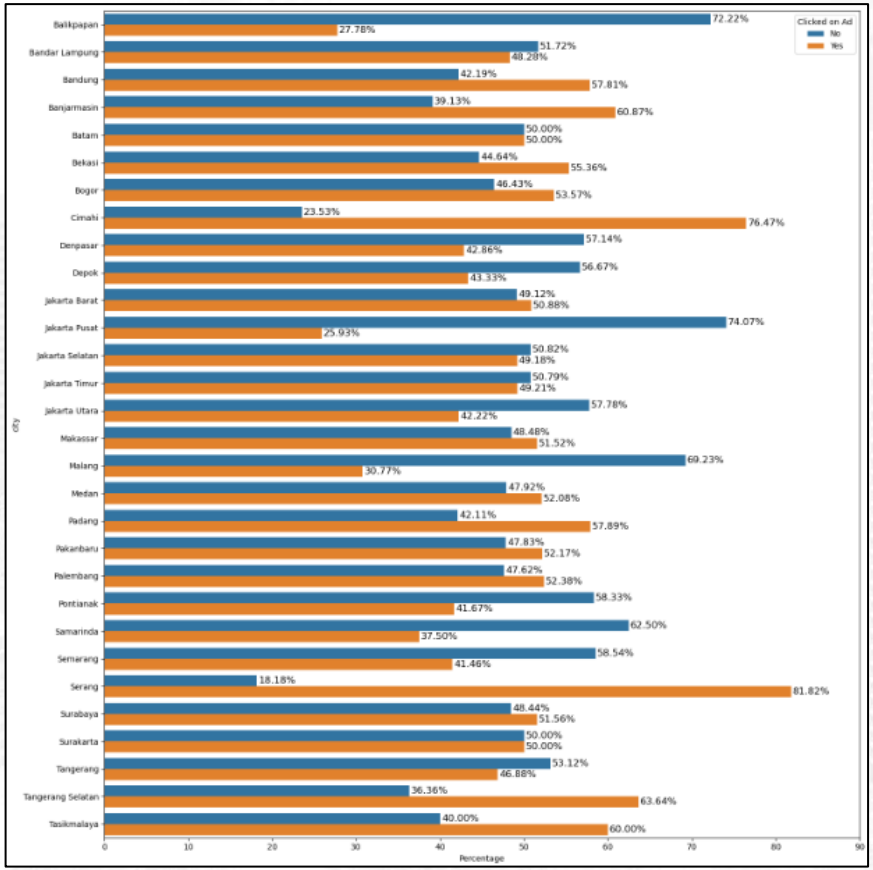
Customer Type and Behaviour Analysis on Advertisement

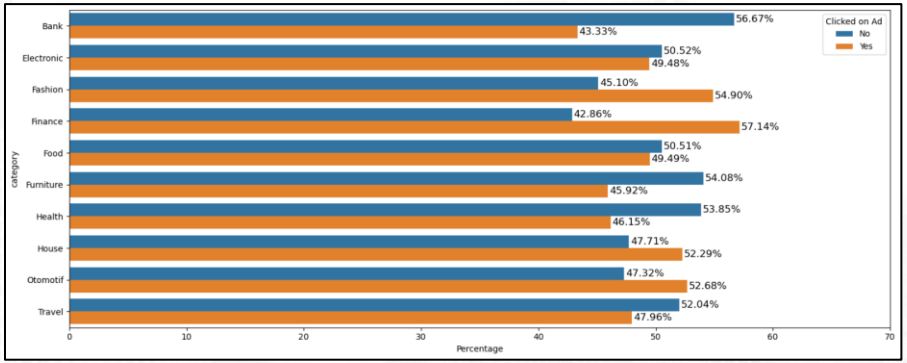


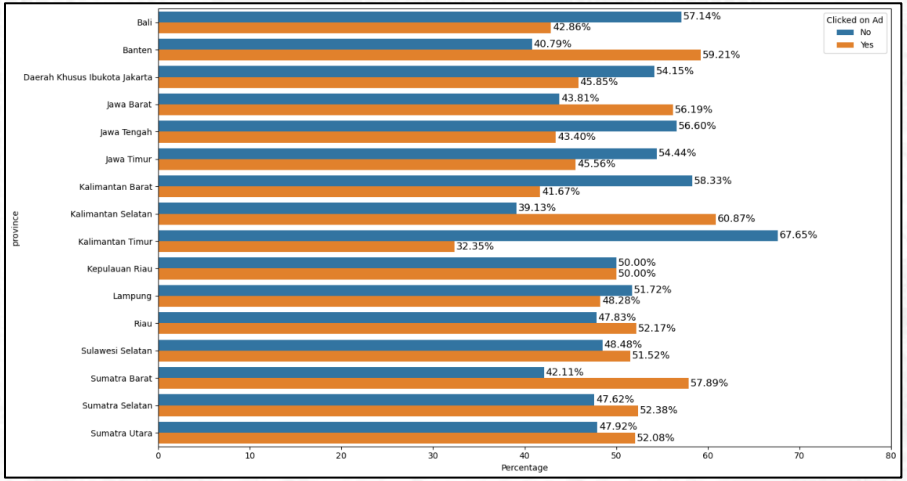
From the bivariate analysis above, we can see that people who clicked tend to have lower daily time spent on site, tend to be older, tend to have less area income, and tend to have lower daily internet usage than those that don't click.

Customer Type and Behaviour Analysis on Advertisement

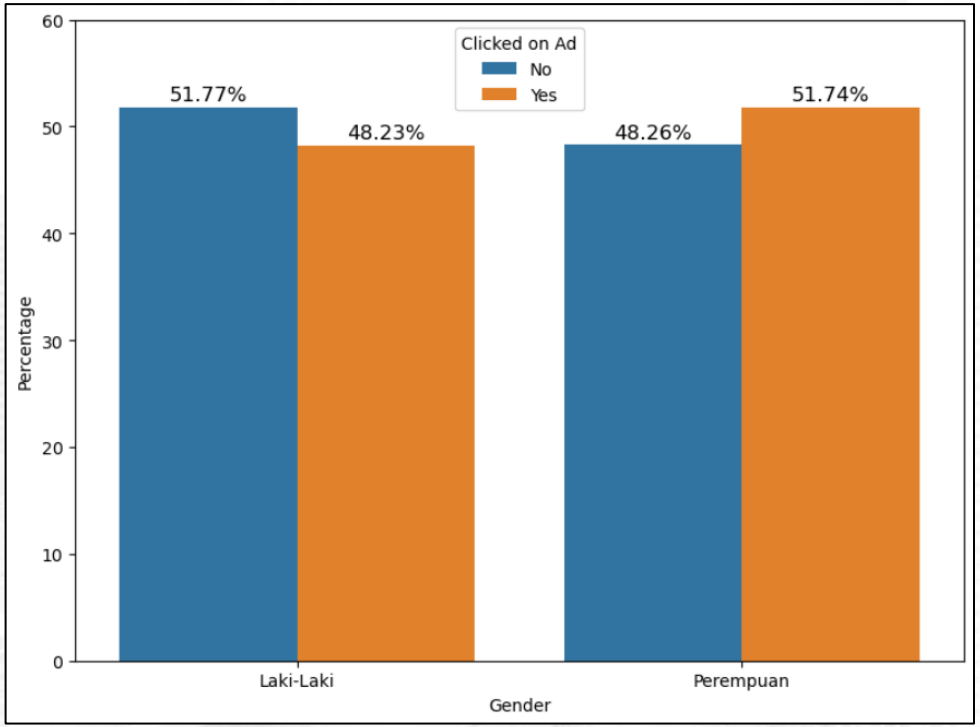
Rakamin Academy





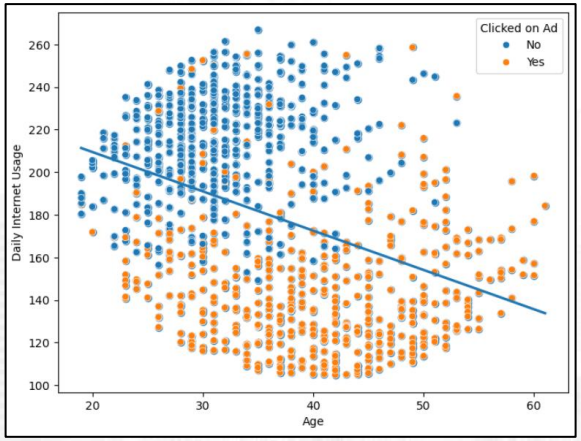


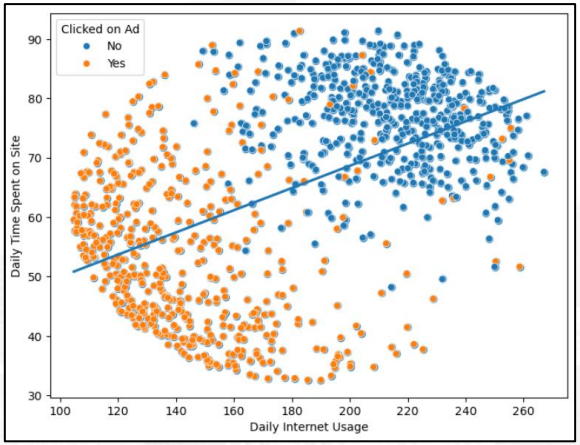
Customer Type and Behaviour Analysis on Advertisement

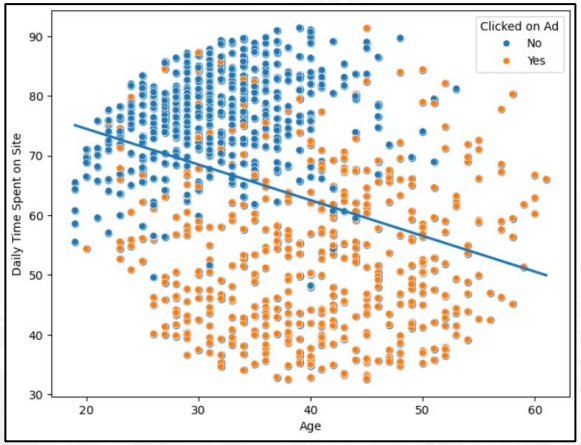


From the countplot to the left and the previous slide, we can see that females are slightly more likely to click on ads rather than males, that the city with the most click percentage is Serang, and the city with the least click percentage is Jakarta Pusat, that the province with the most click percentage is Kalimantan Selatan, and the province with the least click percentage is Kalimantan Timur, and that the category with the most click percentage is Finance, and the category with the least click percentage is Bank.

Customer Type and Behaviour Analysis on Advertisement

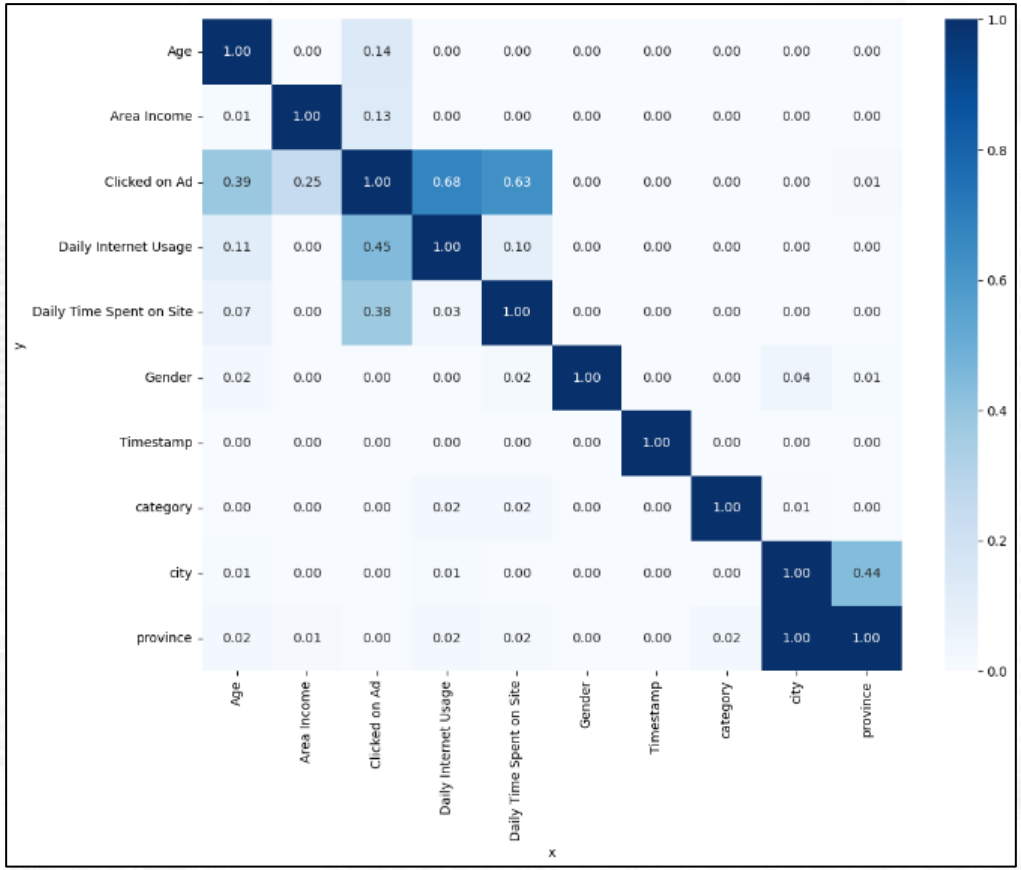






From the scatterplots above, we can see that the correlation between Age and Daily Internet Usage is a negative correlation, that the correlation between Daily Internet Usage and Daily Time Spent on Site is a positive correlation, and that the correlation between Age and Daily Time Spent on Site is a negative correlation.

Customer Type and Behaviour Analysis on Advertisement



From the correlation heatmap to the left, we can see that the columns/features that correlate to the target (Clicked on Ad) are the numerical features. They are Age, Area Income, Daily Internet Usage, and Daily Time Spent on Site.

Data Cleaning & Preprocessing

Data Cleaning & Preprocessing

In this section, we are focusing on the aspect of data preprocessing. The outlines of the processes involved are as follows :

1. NULL value checking & handling
2. Duplicated value checking & handling
3. Feature Engineering
4. Feature Encoding
5. Dataset Splitting

Now, let’s delve deeper into the details of the processes outlined above.

The complete code and .ipynb file of the data preprocessing can be found in the following link.

Data Cleaning & Preprocessing

```
Unnamed: 0                    0
Daily Time Spent on Site     13
Age                           0
Area Income                  13
Daily Internet Usage         11
Gender                        3
Timestamp                     0
Clicked on Ad                 0
city                          0
province                      0
category                      0
dtype: int64
```

There are 4 columns with NULL values. They are 'Daily Time Spent on Site' with 13 NULL values, 'Area Income' with 13 NULL values, 'Daily Internet Usage' with 11 NULL values, and 'Gender' with 3 NULL values.

We fill the numerical column's NULL values with the median value, and the 'Gender' column with the mode value.

```
df.duplicated().sum()
```

```
0
```

As we can see above, there are no duplicates in this dataset.

Data Cleaning & Preprocessing

Rakamin Academy

```
<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  1000 non-null   float64
 2   Age                       1000 non-null   int64
 3   Area Income               1000 non-null   float64
 4   Daily Internet Usage      1000 non-null   float64
 5   Gender                    1000 non-null   object
 6   Timestamp                 1000 non-null   datetime64[ns]
 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: datetime64[ns](1), float64(3), int64(2), object(5)
memory usage: 86.1+ KB
```

The timestamp column contains a date and timestamp. We are going to create new columns/features that will house the year, month, week, and day of the timestamp.

We first have to change the data type of the timestamp column to datetime object. Like shown in the left figure.

day	week	month	year
6	12	3	2016
0	14	4	2016
6	10	3	2016
6	1	1	2016
4	22	6	2016
...	...	...	...
3	6	2	2016
4	16	4	2016
0	5	2	2016
3	12	3	2016
4	22	6	2016

Now that we have successfully changed the data type, we can proceed to the creation of new columns/features like shown in the left figure. We also have to drop the original ‘Timestamp’ column, since it is redundant.

Data Cleaning & Preprocessing

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Unnamed: 0              1000 non-null   int64
 1   Daily Time Spent on Site  1000 non-null   float64
 2   Age                     1000 non-null   int64
 3   Area Income             1000 non-null   float64
 4   Daily Internet Usage    1000 non-null   float64
 5   Gender                  1000 non-null   object
 6   Clicked on Ad           1000 non-null   object
 7   city                    1000 non-null   object
 8   province                1000 non-null   object
 9   category                1000 non-null   object
 10  day                     1000 non-null   int64
 11  week                    1000 non-null   int32
 12  month                   1000 non-null   int64
 13  year                    1000 non-null   int64
dtypes: float64(3), int32(1), int64(5), object(5)
memory usage: 105.6+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Daily Time Spent on Site  1000 non-null   float64
 1   Age                     1000 non-null   int64
 2   Area Income             1000 non-null   float64
 3   Daily Internet Usage    1000 non-null   float64
 4   Gender                  1000 non-null   int64
 5   Clicked on Ad           1000 non-null   int64
 6   day                     1000 non-null   int64
 7   week                    1000 non-null   int32
 8   month                   1000 non-null   int64
 9   category Bank           1000 non-null   uint8
 10  category Electronic     1000 non-null   uint8
 11  category Fashion        1000 non-null   uint8
 12  category Finance        1000 non-null   uint8
 13  category Food           1000 non-null   uint8
 14  category Furniture      1000 non-null   uint8
 15  category Health         1000 non-null   uint8
 16  category House          1000 non-null   uint8
 17  category Otomotif       1000 non-null   uint8
 18  category Travel         1000 non-null   uint8
dtypes: float64(3), int32(1), int64(5), uint8(10)
memory usage: 76.3 KB
```

For the categorical features with binary values, we're going to do a label encoding to it. For the categorical features with over 2 possible values, we're going to do a one-hot encoding. And because the city and province features have to many possible values, we're going to drop them. We also are going to drop the Unnamed column and the year column, since all of the rows have the same year value. The left figure is before encoding, and the right figure is after encoding.

Data Cleaning & Preprocessing

Rakamin Academy

`X_train_scaled.describe()`

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Gender	day	week	month	category Bank	category Electronic	category Fashion	category Finance	category Food	category Furniture	category Health	category House	category Otomotif	category Travel
count	700.000000	700.000000	7.000000e+02	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000
mean	65.073943	36.092857	3.876627e+08	180.335443	0.488571	3.080000	14.301429	3.791429	0.097143	0.107143	0.110000	0.088571	0.105714	0.085714	0.100000	0.108571	0.104286	0.092857
std	15.606714	8.785552	9.301513e+07	43.785808	0.500227	1.973907	8.542697	1.943975	0.296364	0.309516	0.313113	0.284327	0.307691	0.280142	0.300215	0.311323	0.305849	0.290440
min	32.600000	19.000000	1.034285e+08	104.780000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	52.160000	29.000000	3.333581e+08	138.702500	0.000000	1.000000	7.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	68.110000	35.000000	3.990683e+08	182.650000	0.000000	3.000000	14.000000	4.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	78.572500	42.000000	4.607324e+08	219.505000	1.000000	5.000000	22.000000	5.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	91.430000	61.000000	5.563936e+08	267.010000	1.000000	6.000000	29.000000	7.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

`X_train_vanilla.describe()`

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Gender	day	week	month	category Bank	category Electronic	category Fashion	category Finance	category Food	category Furniture	category Health	category House	category Otomotif	category Travel
count	700.000000	700.000000	7.000000e+02	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000	700.000000
mean	65.073943	36.092857	3.876627e+08	180.335443	0.488571	3.080000	14.301429	3.791429	0.097143	0.107143	0.110000	0.088571	0.105714	0.085714	0.100000	0.108571	0.104286	0.092857
std	15.606714	8.785552	9.301513e+07	43.785808	0.500227	1.973907	8.542697	1.943975	0.296364	0.309516	0.313113	0.284327	0.307691	0.280142	0.300215	0.311323	0.305849	0.290440
min	32.600000	19.000000	1.034285e+08	104.780000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	52.160000	29.000000	3.333581e+08	138.702500	0.000000	1.000000	7.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	68.110000	35.000000	3.990683e+08	182.650000	0.000000	3.000000	14.000000	4.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	78.572500	42.000000	4.607324e+08	219.505000	1.000000	5.000000	22.000000	5.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	91.430000	61.000000	5.563936e+08	267.010000	1.000000	6.000000	29.000000	7.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

`X_test_scaled.describe()`

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Gender	day	week	month	category Bank	category Electronic	category Fashion	category Finance	category Food	category Furniture	category Health	category House	category Otomotif	category Travel
count	300.000000	300.000000	3.000000e+02	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000
mean	64.730367	35.813333	3.789515e+08	178.864867	0.456667	3.026667	14.663333	3.876667	0.073333	0.073333	0.083333	0.096667	0.083333	0.126667	0.113333	0.110000	0.130000	0.110000
std	16.088139	8.797147	9.442953e+07	43.315902	0.498951	2.057523	8.153420	1.888565	0.261118	0.261118	0.276847	0.295997	0.276847	0.333155	0.317529	0.313413	0.336865	0.313413
min	32.910000	19.000000	9.797550e+07	105.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	49.952500	29.000000	3.204564e+08	139.395000	0.000000	1.000000	8.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	67.825000	35.000000	3.967774e+08	182.425000	0.000000	3.000000	15.000000	4.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	77.905000	42.000000	4.471783e+08	216.645000	1.000000	5.000000	22.000000	5.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	90.970000	59.000000	5.553263e+08	258.260000	1.000000	6.000000	29.000000	7.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

`X_test_vanilla.describe()`

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Gender	day	week	month	category Bank	category Electronic	category Fashion	category Finance	category Food	category Furniture	category Health	category House	category Otomotif	category Travel
count	300.000000	300.000000	3.000000e+02	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000	300.000000
mean	64.730367	35.813333	3.789515e+08	178.864867	0.456667	3.026667	14.663333	3.876667	0.073333	0.073333	0.083333	0.096667	0.083333	0.126667	0.113333	0.110000	0.130000	0.110000
std	16.088139	8.797147	9.442953e+07	43.315902	0.498951	2.057523	8.153420	1.888565	0.261118	0.261118	0.276847	0.295997	0.276847	0.333155	0.317529	0.313413	0.336865	0.313413
min	32.910000	19.000000	9.797550e+07	105.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	49.952500	29.000000	3.204564e+08	139.395000	0.000000	1.000000	8.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	67.825000	35.000000	3.967774e+08	182.425000	0.000000	3.000000	15.000000	4.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	77.905000	42.000000	4.471783e+08	216.645000	1.000000	5.000000	22.000000	5.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	90.970000	59.000000	5.553263e+08	258.260000	1.000000	6.000000	29.000000	7.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

We are going to split the dataset into train and test with the training dataset percentage of 0.7 and testing dataset of 0.3. We also are going to make one more copy of the split dataset, to compare later on between a scaled/normalized dataset, with a vanilla one.

The top figure is the X_train for the scaled and vanilla version, while the bottom figure is the X_test for the scaled and vanilla version.

```
y_train.value_counts()
```

```
Clicked on Ad
0                350
1                350
dtype: int64
```

```
y_test.value_counts()
```

```
Clicked on Ad
0                150
1                150
dtype: int64
```

As we can see in the figures to the left and in the previous slide, the features and target have ben successfully split, and the target proportions are the same as the original proportion of 50:50.

Data Modeling

Data Modeling

In this section, we are going to focus on the aspect of modelling. The outlines of the processes involved are as follows :

1. Splitting the dataset into a scaled and vanilla versions for experimental purposes.
2. Modelling each dataset versions with 5 different model types.
3. Tuning the hyperparameters of the said models from the said dataset versions.
4. Model selection and evaluation, including the learning curves, confusion matrix, and feature importances.

Now, let’s delve deeper into the details of the outlined processes above.

The complete code and .ipynb file of the data modelling can be found in the following link.

Data Modeling

We're going to try 5 different types of models. They are :

1. Logistic Regression (Linear Model)
2. K-Nearest Neighbor (Neighbor Model)
3. Decision Tree (Tree Based Model)
4. Random Forest (Bagging Algorithm)
5. XGBoost (Boosting Algorithm)

Because the target classes are perfectly balanced, our main performance metric will be the accuracy score. With the recall and precision scores as complementary metrics.

Data Modeling

Rakamin Academy

Accuracy scores before hyperparameter tuning :

Dataset Type	Logistic Regression	KNN	Decision Tree	Random Forest	XGBoost
Vanilla	0.5	0.67	0.93	0.96	0.96
Scaled	0.97	0.93	0.94	0.96	0.96

Accuracy scores after hyperparameter tuning :

Dataset Type	Logistic Regression	KNN	Decision Tree	Random Forest	XGBoost
Vanilla	0.96	0.71	0.96	0.96	0.95
Scaled	0.97	0.93	0.96	0.96	0.95

Data Modeling

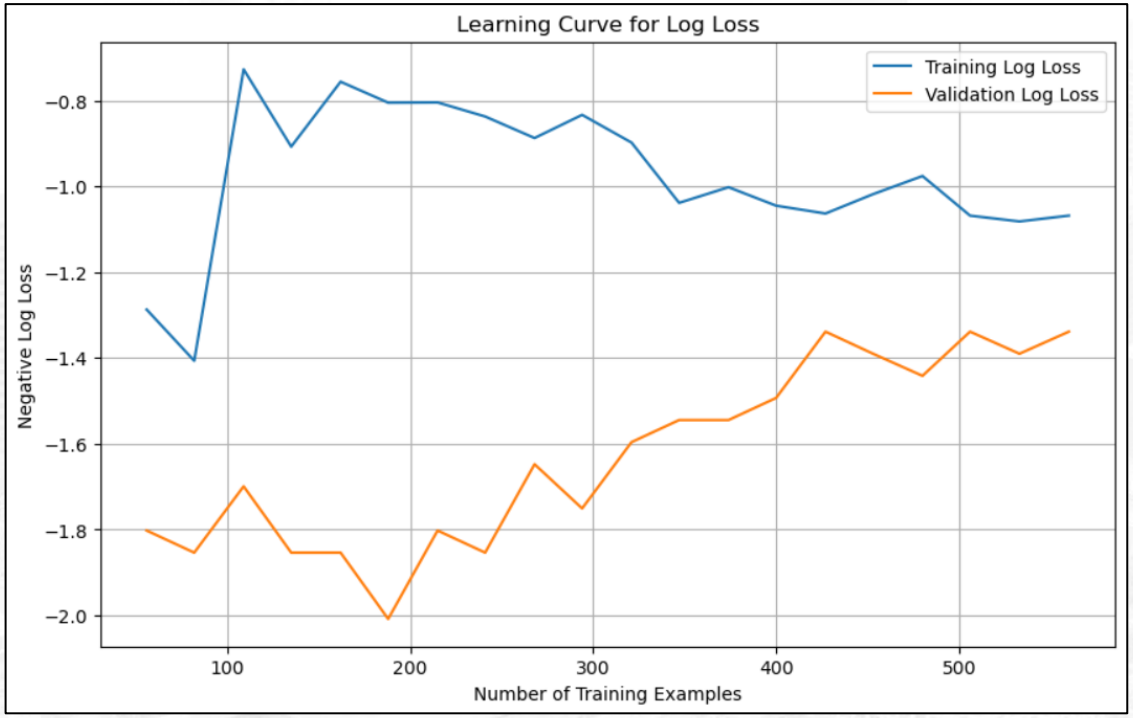
Before the hyperparameter tuning, the logistic regression & knn model's results with the vanilla dataset are very poor to say the least. After the hyperparameter tuning, the logistic regression model's performances sky-rocketed when the hyperparameters are tuned. Apparently when we change the solver from the default 'lbfgs' to 'liblinear', the results improve massively. Then if we look at the other models, they all are either the same or have a slight increase in performance. With KNN still underperforming when compared to the rest.

In the scaled dataset version, there are fewer differences. The only notable difference is that the decision tree model is slightly better after the hyperparameter tuning. The rest is relatively the same. However, one interesting note is that the optimal hyperparameters of the tree-based models (decision tree, random forest, and xgboost) are the same between the vanilla and the scaled dataset. This is interesting and expected as well. Since tree-based models don't use distance to predict, rather they literally use a tree of decisions to classify the datasets. And that is why the optimal hyperparameters in the tree-based models are the same, but not with the knn and logistic regression models.

From all the models and dataset variation (vanilla & scaled), the winner model by a slight margin is the logistic regression model from the scaled dataset. Not only is it the best performing model, but logistic regression models are also simple, computationally cheap, and relatively interpretable compared to its counterparts.

From the cross-validated performance metrics on the scaled tuned logistic regression, it seems that the model is a good fit. But, let’s look at the log loss learning curve to be sure.

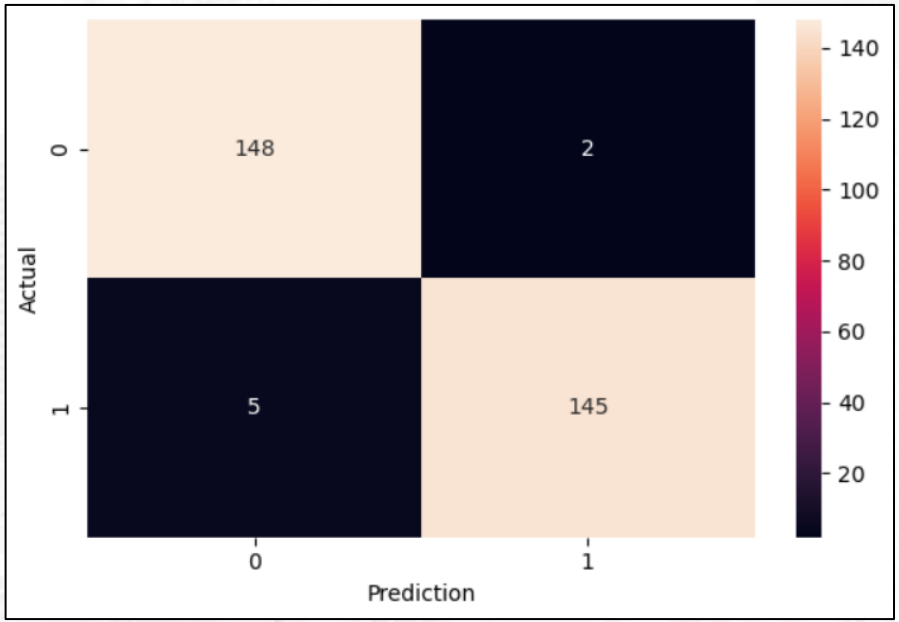
Data Modeling



We can see that with more training samples, the training and validation sets eventually converge. This means that the model is a good fit.

Therefore, we are going to choose the logistic regression model with the scaled dataset.

Data Modeling



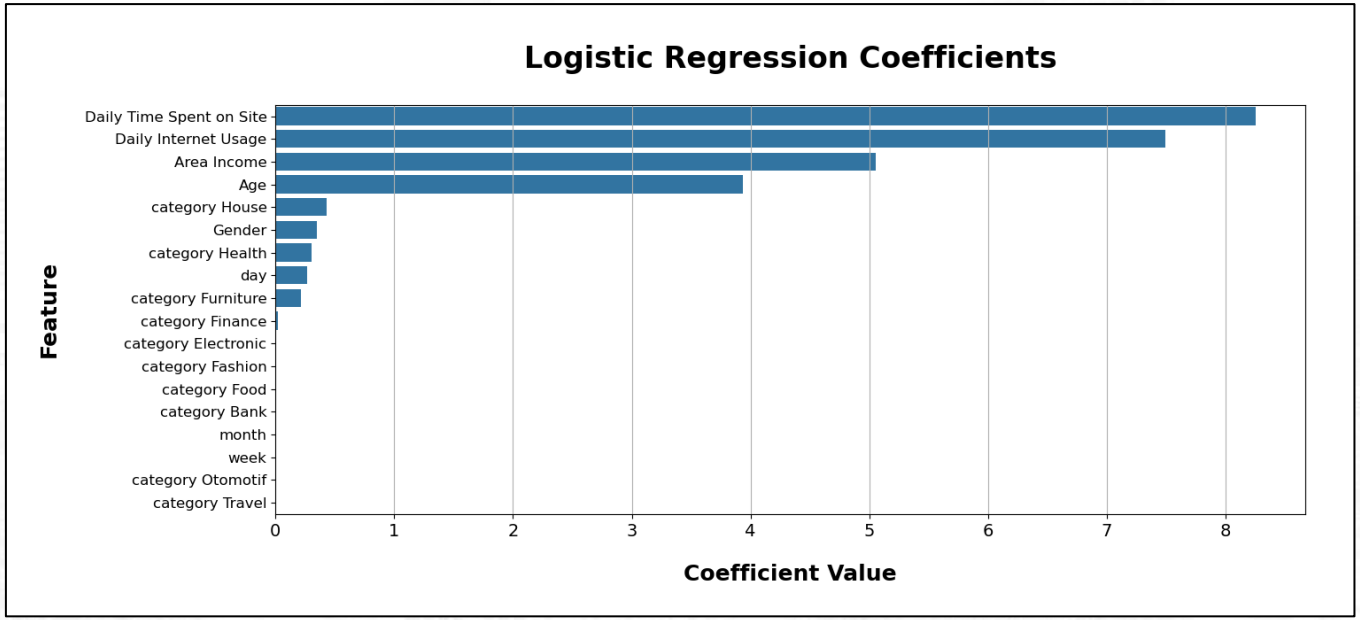
Very solid and amazing results, when considering that the model is a good fit.

From the confusion matrix to the left, we can calculate the accuracy, recall, and precision. And they are as follows :

1. **Accuracy** : (True Positive + True Negative) / Total. (148 + 145) / 300 = **0.977**
2. **Recall** : True Positive / (True Positive + False Negative). 145 / (145 + 5) = **0.967**
3. **Precision** : True Positive / (True Positive + False Positive). 145 / (145 + 2) = **0.986**

Data Modeling

	Feature	Coefficient
0	Daily Time Spent on Site	8.256890
1	Daily Internet Usage	7.494279
2	Area Income	5.054257
3	Age	3.937822
4	category House	0.433807
5	Gender	0.345201
6	category Health	0.306026
7	day	0.267409
8	category Furniture	0.215671
9	category Finance	0.019722
10	category Electronic	0.007737
11	category Fashion	0.000000
12	category Food	0.000000
13	category Bank	0.000000
14	month	0.000000
15	week	0.000000
16	category Otomotif	0.000000
17	category Travel	0.000000



From the coefficient dataframe and its barplot above, we can see that the most important features are 'Daily Time Spent on Site', 'Daily Internet Usage', 'Area Income' & 'Age'. Other features also play a smaller role, and they are 'category House', 'Gender', 'category Health', 'day', and 'category Furniture'. The rest of the features have negligable influences to the model.

Business Recommendation & Simulation

Business Recommendation & Simulation

In this section, we will focus on the aspect of business recommendation and simulation. The outlines of the processes involved are as follows :

1. Business recommendation making.
2. Business simulation making (before modelling).
3. Business simulation making (after modelling).

Now, let us go into the details of the processes outlined above.

The complete code and .ipynb file of the business recommendation & simulation can be found in the following link.

Business Recommendation & Simulation

	Feature	Coefficient
0	Age	3.937822
1	category House	0.433807
2	day	0.267409
3	category Finance	0.019722
4	category Travel	0.000000
5	category Otomotif	0.000000
6	category Food	0.000000
7	week	0.000000
8	month	0.000000
9	category Bank	0.000000
10	category Fashion	0.000000
11	category Electronic	-0.007737
12	category Furniture	-0.215671
13	category Health	-0.306026
14	Gender	-0.345201
15	Area Income	-5.054257
16	Daily Internet Usage	-7.494279
17	Daily Time Spent on Site	-8.256890

Looking at the feature importance to the left, we can make several business recommendations. And they are as follows :

1. Target older customers, since the relationship of the feature 'Age' is highly positive with the target.
2. Send out the ads on later days of the week (week-end or near week-end), since the relationship of the feature 'day' is positive with the target.
3. Maximize advertisements on house and finance categories, since they have positive relationship with the target.
4. Avoid advertising about electronic, furniture, and health categories, since they have negative relationship with the target.
5. Target female customers more, since the relationship of the feature 'Gender' is negative with the target (and the female gender is encoded as 0).
6. Target people with small income, since the relationship of the feature 'Area Income' is highly negative with the target.
7. Target people with low daily internet usage and low daily time spent on site, since the relationship of the feature 'Daily Internet Usage' & 'Daily Time Spent on Site' is highly negative with the target.

Before we make the simulation, we first need to address an assumption. That assumption is the cost to contact a customer (or deliver an ad to a customer), and the revenue that they bring if they click/accept the advertisement. For simplicity, let's assume that the cost to contact a customer is 1000 rupiah per customer, and the revenue that they will bring if they click on the ad is 4000 rupiah. With that being said, we can now delve deeper into the business simulation.

Before the modelling, an advertisement is sent to each and every one of the customers, and only 50% of them clicked on the advertisement. If we calculate the return of investment, that would be total revenue divided by total contact cost, or ((4,000 x 500 (amount of people that clicked on the ad)) / (1,000 x 1,000 (amount of customers contacted))) x 100% = 200%. The **return of investment** is **200%**, or twice the cost to contact (2,000,000 rupiahs).

Business Recommendation & Simulation

Rakamin Academy

Now, **after the modelling**, we only send advertisements to customers that we predict will click on the ad based on the model that we have developed. The test set consists of 300 customers. And according to the model, there are only 147 people that will click on the ad. If we contact all of them, then the total contact cost will be (147 x 1,000) = 147,000 rupiahs. And of all that customers, 2 of them are false positives. Meaning that the model predicts that those 2 people will click on the ad, but in reality they won't. That means, the revenue will be (145 x 4,000) = 580,000 rupiahs. And the **return of investment** will be (580,000 / 147,000) x 100% = **394.56%**. Which is a **97.28%** increase from before the modelling. We also need to consider the opportunity loss, which basically is the false negative. Opportunity loss is the amount of people that our model predict won't click on the ad, but in reality they actually will click. And if we look at the confusion matrix, there are 5 of them. This means that the **opportunity loss** is (revenue x total false negatives) or (4,000 x 5) = 20,000 rupiahs. Which is only **3.45%** of the total revenue.

Therefore in **conclusion**, we can see that after the modelling, the return of investment increases by **97.28%** compared to before the modelling. And that is achieved with a relatively negligible amount of opportunity loss.