Predict Customer Clicked Ads Classification by Using Machine Learning

Created by:
Indra Maulidin
indramaulidin@gmail.com
linkedin.com/in/indra-maulidin

I'm a Data Enthusiast and have a desire to create a career in Data Field especially as Data Scientist or Data Analyst. I have skills to use programming language such as Python and SQL. I have project experience doing Exploratory Data Analysis, Data Pre-Processing and creating Machine Learning Model.

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

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 ”

Descriptive Statistics

```
<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
```

	Gender	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

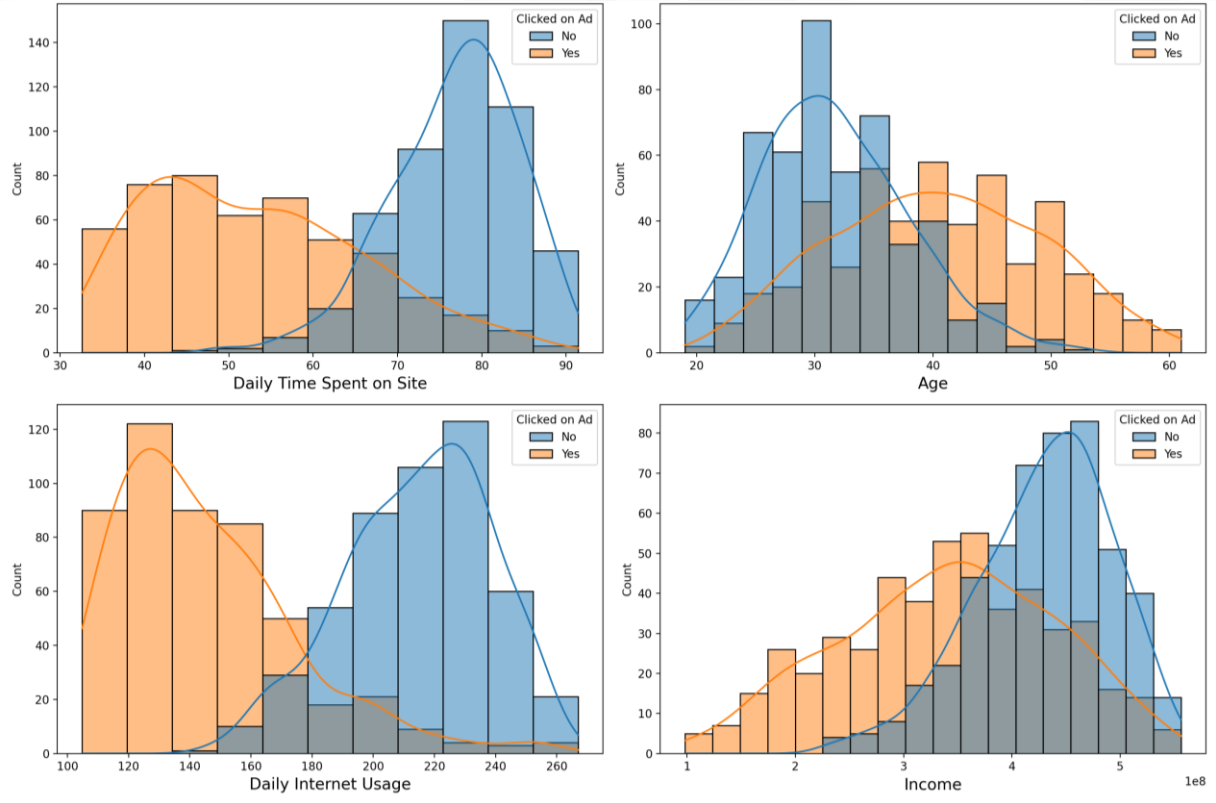
	Index	Daily Time Spent on Site	Age	Income	Daily Internet Usage
count	1000.000000	987.000000	1000.000000	9.870000e+02	989.000000
mean	499.500000	64.929524	36.009000	3.848647e+08	179.863620
std	288.819436	15.844699	8.785562	9.407999e+07	43.870142
min	0.000000	32.600000	19.000000	9.797550e+07	104.780000
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

```
df[nums].mean()
```

```
Index                     4.995000e+02
Daily Time Spent on Site  6.492952e+01
Age                       3.600900e+01
Income                    3.848647e+08
Daily Internet Usage      1.798636e+02
dtype: float64
```

Untuk selengkapnya, dapat melihat jupyter notebook disini

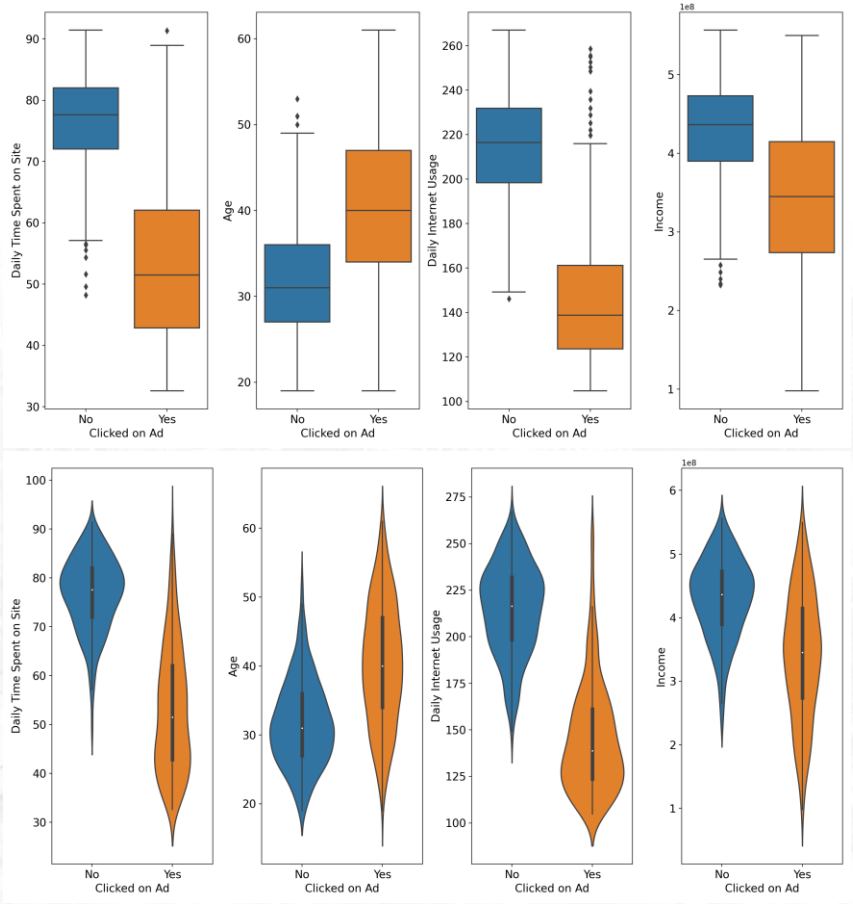
Univariate Analysis



Untuk selengkapnya, dapat melihat jupyter notebook disini

Customer Type and Behaviour Analysis on Advertisement

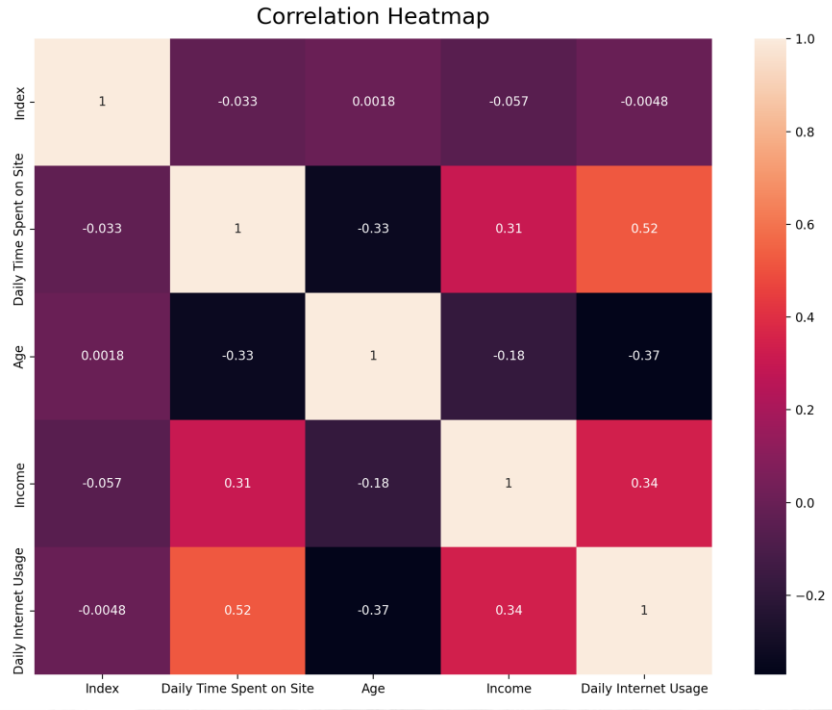
Bivariate Analysis

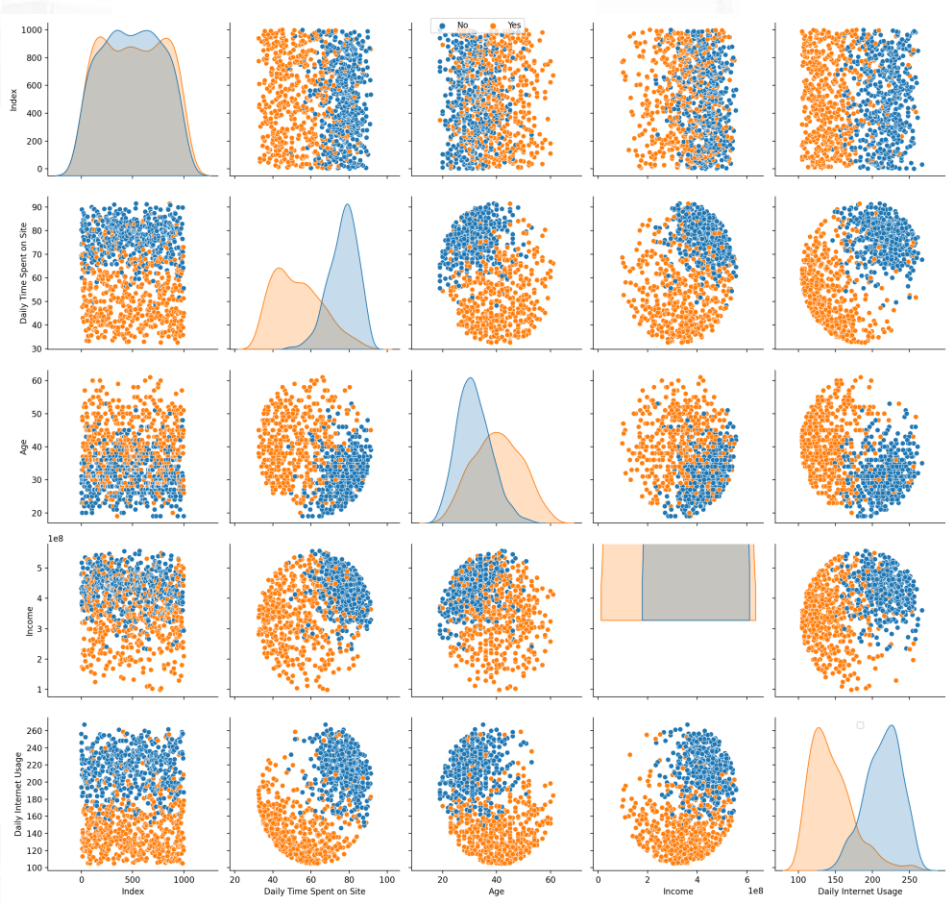


Untuk selengkapnya, dapat melihat jupyter notebook disini

Rakamin Academy

Multivariate Analysis





Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Cleaning & Preprocessing

```
df.isnull().sum()
```

Index	0
Daily Time Spent on Site	13
Age	0
Income	13
Daily Internet Usage	11
Gender	3
Timestamp	0
Clicked on Ad	0
City	0
Province	0
Category	0
dtype: int64	

- Terdapat Data Missing Value pada Kolom Daily Time Spent on Site, Income, Daily Internet Usage dan Gender.
- Data Missing Value tersebut diganti dengan Nilai Median atau Modus dari masing – masing Data.

```
[ ]  df_clean['Daily Time Spent on Site'].fillna(df_clean['Daily Time Spent on Site'].median(), inplace=True)
```

```
[ ]  df_clean['Income'].fillna(df_clean['Income'].median(), inplace=True)
```

```
     df_clean['Daily Internet Usage'].fillna(df_clean['Daily Internet Usage'].median(), inplace=True)
```

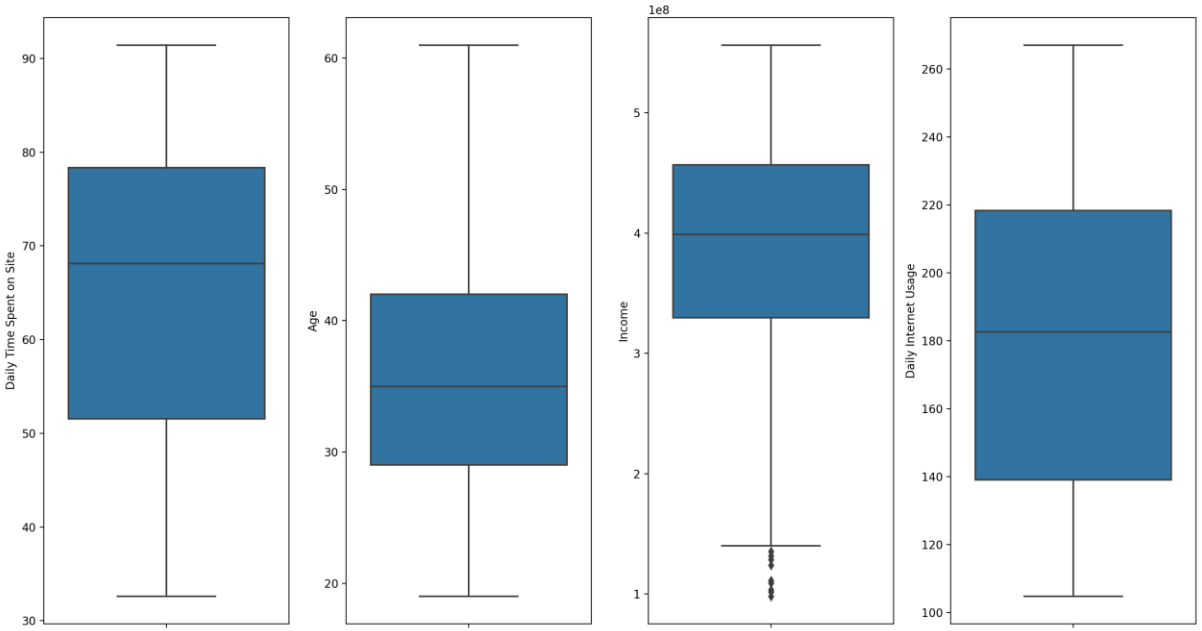
```
[ ]  df_clean['Gender'].fillna(df_clean['Gender'].mode()[0], inplace=True)
```

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Cleaning & Preprocessing

```
df_clean.duplicated().sum()
0
```

- Dataset tidak memiliki Duplicated Data.



- Pada Kolom Income Terdapat Outlier.
- Maka kita perlu mengatasi Outlier tersebut sebelum melakukan Modeling.
- Outlier pada Kolom Income diatasi dengan menggunakan IQR.

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Cleaning & Preprocessing

Rakamin Academy

Label Encoding

```
[33] # Feature Gender
     mapping_gender = {
         'Perempuan' : 0,
         'Laki-Laki' : 1
     }
     df_preprocessing['Gender'] = df_preprocessing['Gender'].map(mapping_gender)
```

```
[34] # Feature Clicked on Ad
     mapping_ad = {
         'No' : 0,
         'Yes' : 1
     }
     df_preprocessing['Clicked on Ad'] = df_preprocessing['Clicked on Ad'].map(mapping_ad)
```

- Dilakukan Label Encoding untuk Kolom Gender dan Clicked on Ad.
- Dilakukan One-hot Encoding untuk Kolom Province dan Category.

One-hot Encoding

```
[35] prov = pd.get_dummies(df_preprocessing['Province'], prefix='Province')
     cat = pd.get_dummies(df_preprocessing['Category'], prefix='Category')
```

```
[36] df_preprocessing = df_preprocessing.join(prov)
     df_preprocessing = df_preprocessing.join(cat)
```

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Cleaning & Preprocessing

Rakamin Academy

Split Dataset to Features and Target

```
[38] X = df_preprocessing.drop(columns = ['Index','Clicked on Ad','City','Province','Category'])
     y = df_preprocessing['Clicked on Ad']
```

Feature Timestamp Extraction

```
[43] X['Timestamp'] = pd.to_datetime(X['Timestamp'])
```

```
[45] from datetime import date as dt
```

```
[46] X['Year'] = X['Timestamp'].dt.year
     X['Month_Number'] = X['Timestamp'].dt.month
     X['Week_Number'] = X['Timestamp'].dt.isocalendar().week
     X['Day_Number'] = X['Timestamp'].dt.day
```

```
[47] X = X.drop(columns='Timestamp')
```

```
[48] X['Week_Number'] = X['Week_Number'].astype('int64')
```

```
[53] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)
```

```
[54] X_train.shape

     (684, 22)
```

```
[55] X_test.shape

     (294, 22)
```

- Data dipisahkan menjadi Features dan Target.
- Dilakukan Feature Extraction menjadi beberapa Feature untuk Feature Timestamp.
- Data dipisahkan menjadi Data Train dan Data Test dengan persentase masing – masing 70% & 30%.

Untuk selengkapnya, dapat melihat jupyter notebook disini

Data Modeling

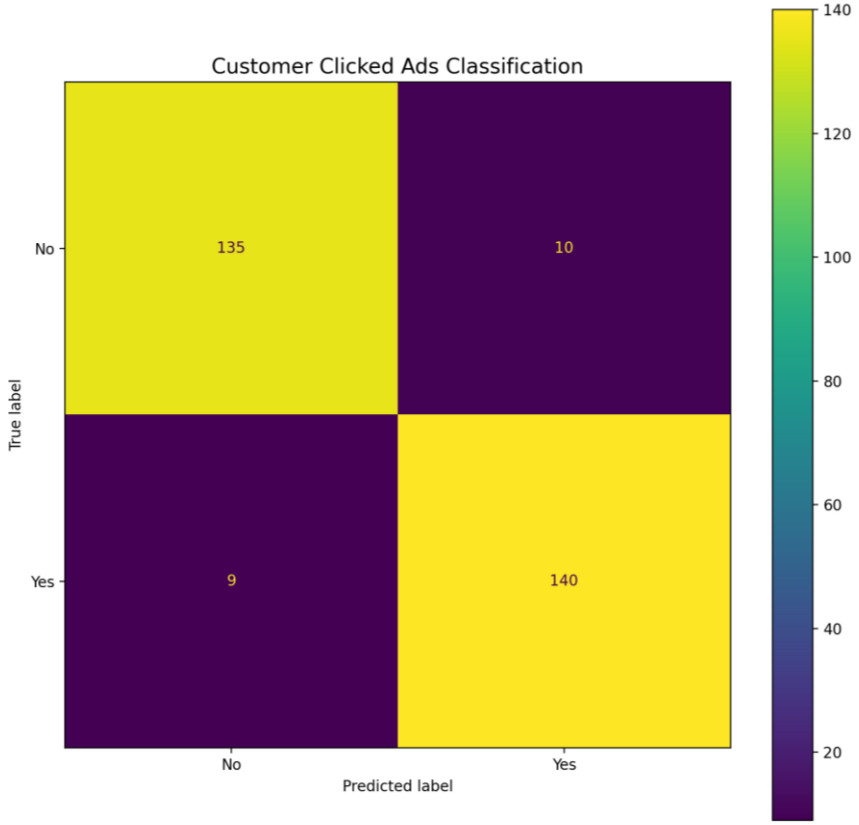
	Experiment	Recall	Accuracy	Precission	F1
0	Random Forest Without Data Scalling	94.63%	93.2%	92.16%	93.38%
1	Random Forest With Data Scalling	93.96%	93.54%	93.33%	93.38%

Terlihat setelah dilakukan Data Scalling performa untuk beberapa Matrix mengalami perubahan :

- Untuk Accuracy dan Precission keduanya mengalami kenaikan masing - masing sebesar 0,34% dan 1,17%.
- Untuk Recall mengalami penurunan sebesar 0,67%.
- Untuk F1 tidak mengalami perubahan.
- Maka untuk analisis selanjutnya akan digunakan Model Random Forest With Data Scalling.
- Karena Dataset memiliki Class yang cukup Balance dan masing-masing Label mempunyai kepentingan yang sama dalam Analisis Bisnis kedepannya, maka kita dapat menggunakan matrix Accuracy dalam menentukan performa Model yang kita buat.

Untuk selengkapnya, dapat melihat jupyter notebook disini

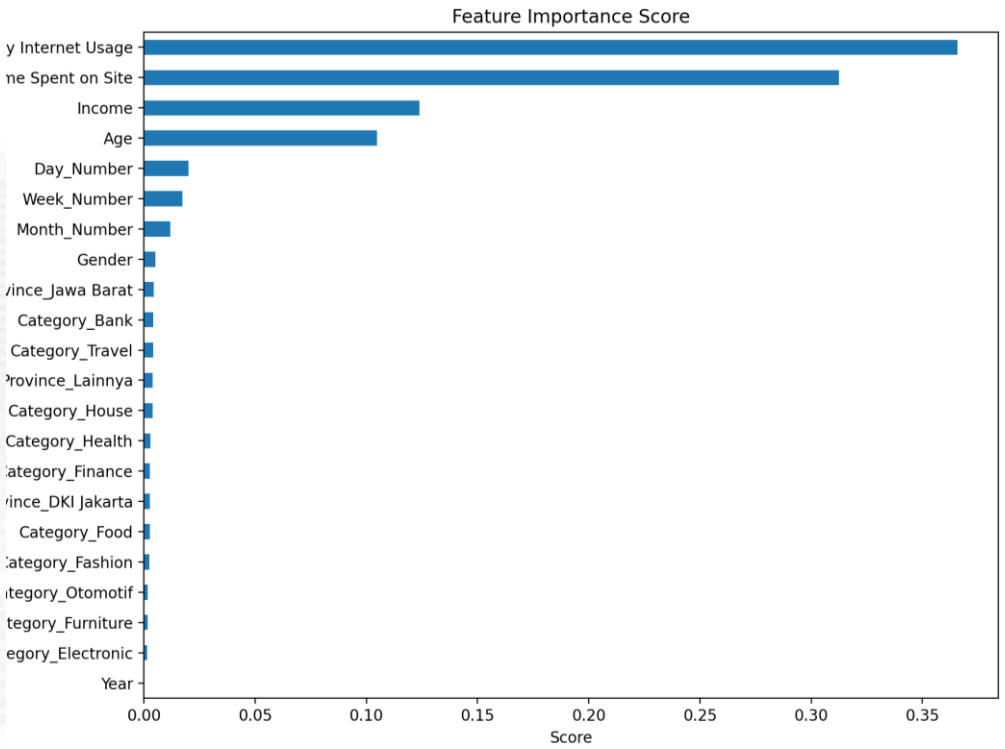
Data Modeling



- Terlihat Model yang dibuat memiliki **Type I Error (False Positive = 10)** atau **Type II Error (False Negative = 9)** yang kecil.
- Karena Nilai Error yang kecil tersebut membuat Model yang dibuat memiliki Accuracy yang tinggi (93,54%) dalam memprediksi Customer yang meng-klik Iklan atau tidak meng-klik Iklan.

Untuk selengkapnya, dapat melihat jupyter notebook disini

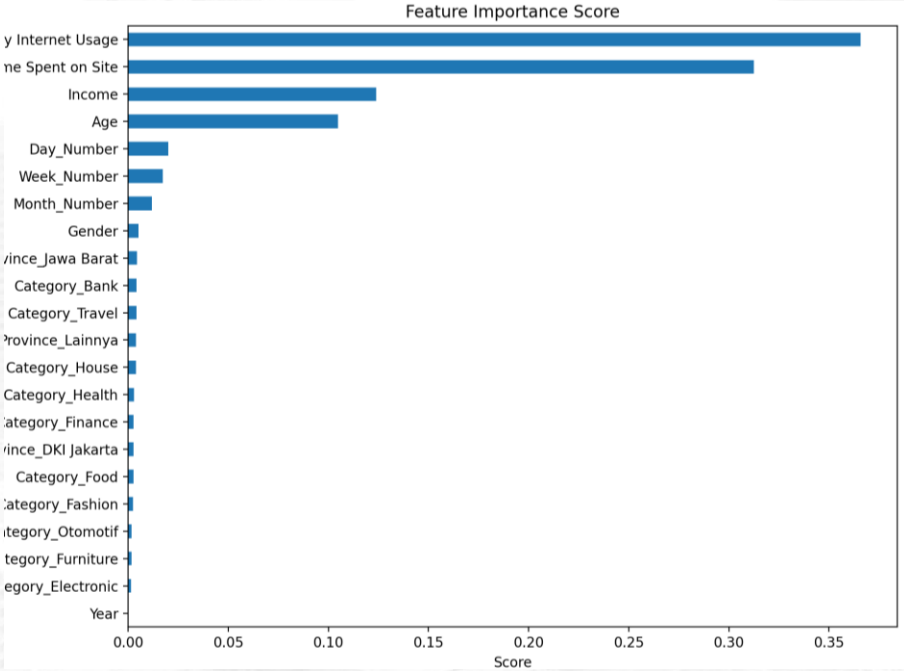
Data Modeling



- Untuk Feature **Daily Internet Usage** dan **Daily Time Spent on Site** keduanya memiliki Score Importances yang tinggi dibandingkan dengen Feature Lainnya.
- Kedua Feature tersebut dapat digunakan sebagai Feature Utama dalam menentukan keberhasilan Marketing kedepannya.
- Tim Marketing juga dapat memper timbangkan **Feature Income** dan **A ge** sebagai Feature Tambahan dal am menentukan Strategi Marketin g kedepannya.

Untuk selengkapnya, dapat melihat jupyter notebook disini

Business Recommendation & Simulation



- Untuk Feature Daily Internet Usage dan Daily Time Spent on Site keduanya memiliki Score Importances yang tinggi dibandingkan dengen Feature Lainnya.
- Kedua Feature tersebut dapat digunakan sebagai Feature Utama dalam menentukan keberhasilan Marketing kedepannya.
- Berdasarkan EDA yang telah dilakukan, Kelompok Customer dengan Daily Internet Usage 100 - 125, Daily Time Spent on Site 30 - 40 dan Umur > 40 Tahun lebih memungkinkan untuk meng-klik Iklan.
- Kita dapat melakukan analisis lebih lanjut terhadap perilaku kelompok customer tersebut sehingga kita dapat mengoptimasikan campaign yang diberikan kepada kelompok customer tersebut. Dengan melakukan optimasi tersebut diharapkan kita dapat mengurangi kerugian yang diakibatkan cost yang dikeluarkan perusahaan.
- Tim Marketing juga dapat mempertimbangkan Feature Income dan Age sebagai Feature Tambahan dalam menentukan Strategi Marketing kedepannya.

Untuk selengkapnya, dapat melihat jupyter notebook disini

Skema Bisnis

- Diasumsikan perusahaan menjalankan Bisnis Digital Marketing dengan Cost Rp 10.000 untuk setiap Customer.
- Perusahaan akan mendapatkan keuntungan berdasarkan Jumlah Customer yang Convert dengan biaya Rp 13.000 per Customer.
- Disini Kita akan Fokus terhadap **kerugian** yang diakibatkan **Cost yang sudah dikeluarkan** perusahaan untuk menayangkan Iklan tetapi **tidak menghasilkan Revenue** bagi perusahaan (**Customer yang tidak mengklik Iklan**).

Untuk selengkapnya, dapat melihat jupyter notebook disini

Business Recommendation & Simulation

Actual Label \ Predicted Label	Not Clicked Ads	Clicked Ads
Jumlah Sample Test = 294		
Not Clicked Ads	135	10
Clicked Ads	9	140

- Secara keseluruhan dari 294 Sample Customer, Machine Learning akan memprediksi **144 Customer tidak mengklik iklan** dan 150 Customer mengklik iklan atau sebesar **49%** dan **51%**.
- Berdasarkan Tabel Confussion Matrix disamping, Machine Learning yang dibuat berhasil **memprediksi dengan tepat** Customer yang **tidak mengklik Iklan** sebesar **135 Customer** dari Total 294 Sample Customer atau sekitar **46%**.
- Dari **150 Customer** yang diprediksi akan **mengklik Iklan** oleh Machine Learning**,** **140 Customer diprediksi dengan tepat** atau sekitar **93%**. Sedangkan sisanya sekitar 7 % merupakan Error.

Untuk selengkapnya, dapat melihat jupyter notebook disini

Business Recommendation & Simulation

Skema Tanpa Machine Learning

Data berdasarkan Dataset

Terdapat 1000 Customer

500 Customer Tidak Mengklik Iklan (50%)

500 Customer Mengklik Iklan (50%)

Revenue = 500 x 13.000 = 6.500.000

Marketing Cost = 1000 x 10.000 = 10.000.000

Skema Dengan Machine Learning

Data berdasarkan Dataset

Terdapat 1000 Customer

490 Customer Tidak Mengklik Iklan (49%)

510 Customer Mengklik Iklan (51%)

474 Customer benar Mengklik Iklan (93% dari 510).

Revenue = 474 x 13.000 = 6.162.000

Marketing Cost = 510 x 10.000 = 5.100.000

Rp	Tanpa Machine Learning	Dengan Machine Learning
Revenue	6.500.000	6.162.000
Marketing Cost	10.000.000	5.100.000
Profit	- 3.500.000	1.062.000

Rp

1.062.000

Profit With Machine Learning

Kesimpulan

- Saat Perusahaan melakukan **Marketing tanpa menggunakan Machine Learning**, Perusahaan akan mendapatkan **Potential Loss sebesar Rp 3.500.000**.
- Sedangkan jika Perusahaan melakukan **Marketing dengan menggunakan Machine Learning**, Perusahaan akan mendapatkan **Potential Revenue sebesar Rp 1.062.000**.
- Terlihat **dengan menggunakan Machine Learning** Perusahaan dapat **mengurangi Cost** yang dikeluarkan **sebesar 49 %**, **tetapi** Perusahaan akan mengalami **penurunan Revenue sebesar 5,2%**.
- Kedepannya perlu dilakukan **Analisis Evaluasi Model Lebih lanjut** untuk **meningkatkan Revenue** Perusahaan **tanpa meningkatkan Cost** dan **tanpa mengurangi Profit yang cukup besar**.

Untuk selengkapnya, dapat melihat jupyter notebook disini