Rakamin Academy

Predict Clicked Ads Visitor Classification by using Machine Learning

Created by:

Muhammad Ariq Arfina

ariqarfina05@gmail.com

LinkedIn : Muhammad Ariq Arfina

Github : ariqarfina

Supported by:

Rakamin Academy

Career Acceleration School

www.rakamin.com

Overview

“A company in Indonesia wants to know the effectiveness of an advertisement that they run. this is important for the company because it allows them to determine how successful the advertisements are in attracting Visitors to see advertisements. It can help companies determine marketing targets by processing historical advertisement data and finding insights and patterns that occur. The focus of this case is to create machine learning classification models that function to determine the right target Visitors.”

Business Metrics :

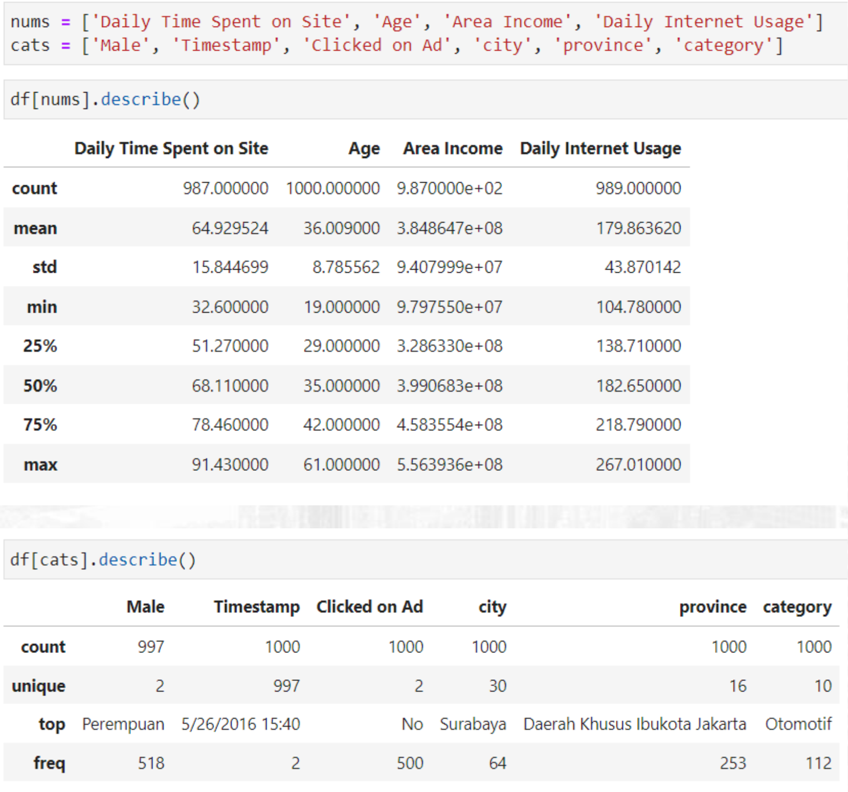
- Advertisement Cost
- Engagement Rate

Dataset Info

Feature	Details
Unnamed: 0	Index
Daily Time Spent on Site	Shows the daily time spent on the site
Age	Visitor's Age
Area Income	Visitor's Income
Daily Internet Usage	the amount of time spent on the internet in a singl e day
Male	Gender
Timestamp	Time
Clicked on Ad	Did the Visitor click on the ad?
City	Visitor's city of residence
province	Visitor's province of residence
category	Ad category

For more details, you can see Dataset here

Statistical Analysis



```
nums = ['Daily Time Spent on Site', 'Age', 'Area Income', 'Daily Internet Usage']
cats = ['Male', 'Timestamp', 'Clicked on Ad', 'city', 'province', 'category']
```

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

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

```
df[cats].describe()
```

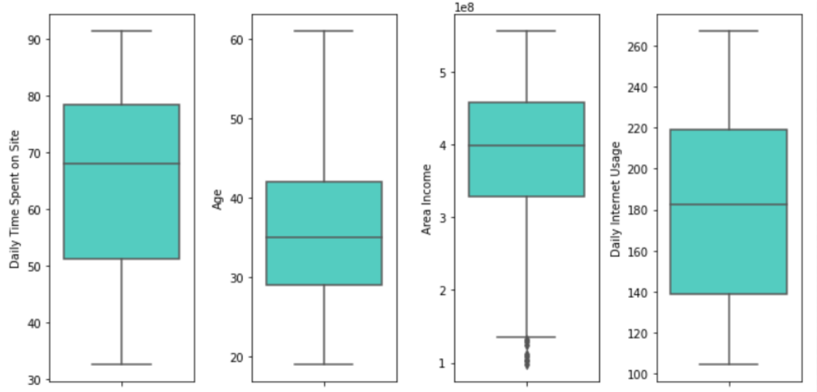
	Male	Timestamp	Clicked on Ad	city	province	category
count	997	1000	1000	1000	1000	1000
unique	2	997	2	30	16	10
top	Perempuan	5/26/2016 15:40	No	Surabaya	Daerah Khusus Ibukota Jakarta	Otomotif
freq	518	2	500	64	253	112

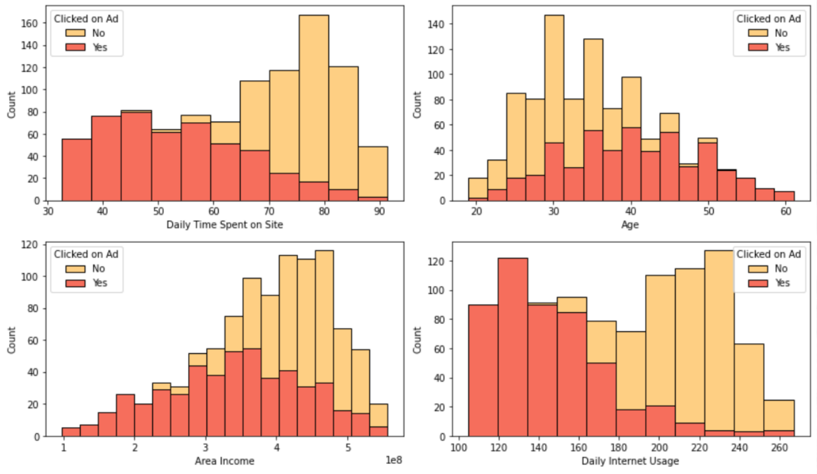
Observation:

1. The columns with symmetrical distributions are `**Daily Time Spent on Site**`, and `**Daily Internet Usage**`.
2. `**Age**` and `**Area Income**` appears to be skewed.
3. Women are the most common Visitors in Surabaya, but the province is most prevalent in DKI Jakarta. It can be ambiguous, so we must take action in this case.
4. The most `**category**` clicked is Otomotif, and the distribution of `**clicked on ads**` is normally distributed

For more details, you can see Jupiter Notebook here

Univariate Analysis



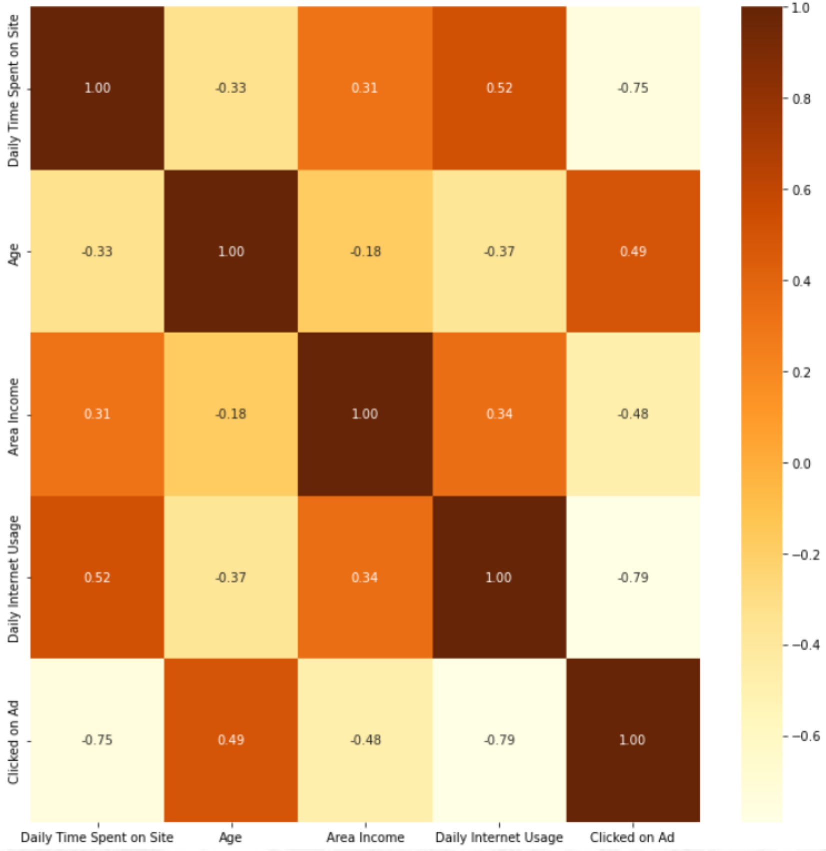


Observation:

1. **\`Area Income\`** column looks more skewed to right. The **\`Age\`** column looks skewed to left
2. **\`Area Income\`** has outlier data
3. **\`Daily Time Spent on Site\`** and **\`Daily Internet Usage\`** feature looks bimodal
4. Based on the characteristics listed above, it is clear that **Visitors are divided into several groups**
5. It is divided into three sections based on the **'Age'** feature, namely:
   - Click ads less than 50% = Age 35 and under
   - Click ads less than 50% = Age 35 and under
   - Click ads at 80% or higher = Age 45 and up
6. The distribution of the **'Daily Time Spent on Site'** and **'Daily Internet Usage'** features is closely similar.

For more details, you can see Jupiter Notebook here

Bivariate Analysis



Observation:

1. We can see that **`Age`** and **`Clicked on Ad`** Features have positively correlation
2. **`Daily Internet Usage`** and **`Daily Time Spent on Site`** Features have positively Correlation

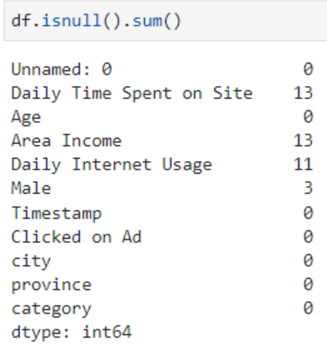
For more details, you can see Jupiter Notebook here

Data Cleaning & Preparation

Rakamin Academy

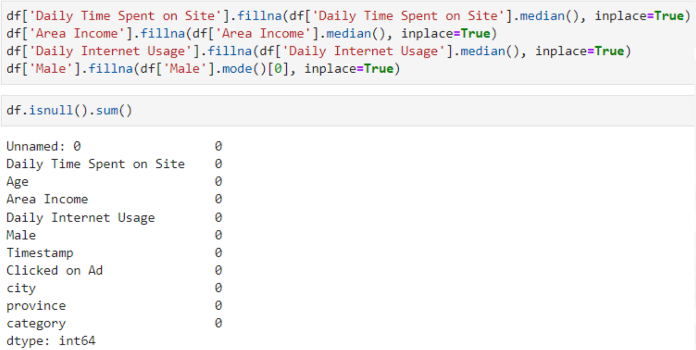
- ## Checking Null Values

Methods for dealing with missing data values include:
1. `**Daily Time Spent on Site**` : Filling null values with median
2. `**Area Income**` : Filling null values with median
3. `**Daily Internet Usage**` : Filling null values with median
4. `**Male**` : Filling null values with mode



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

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



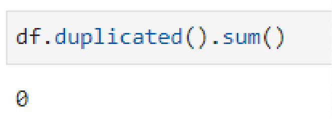
```
df['Daily Time Spent on Site'].fillna(df['Daily Time Spent on Site'].median(), inplace=True)
df['Area Income'].fillna(df['Area Income'].median(), inplace=True)
df['Daily Internet Usage'].fillna(df['Daily Internet Usage'].median(), inplace=True)
df['Male'].fillna(df['Male'].mode()[0], inplace=True)
```

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

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

- ## Checking Duplicated Values

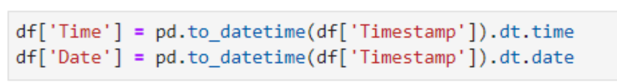
There are no duplicates in the data.



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

0
```

- ## Extract Datetime Data



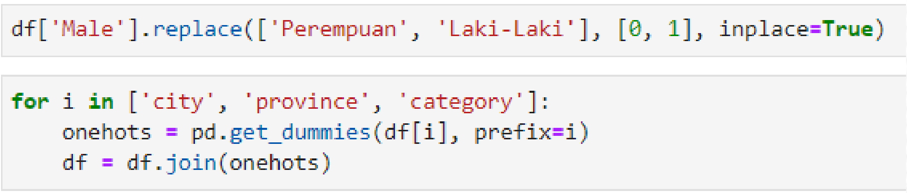
```
df['Time'] = pd.to_datetime(df['Timestamp']).dt.time
df['Date'] = pd.to_datetime(df['Timestamp']).dt.date
```

For more details, you can see Jupiter Notebook here

Data Cleaning & Preparation

- **Feature Encoding**

We need to encoding categorical feature for better insight. Feature that we need to encoding is:
1. `**Male**` : Label Encoding
2. `**Clicked on Ad**` : Label Encoding
3. `**city**` : One Hot Encoding
4. `**province**` : One Hot Encoding
5. `**category**` : One Hot Encoding



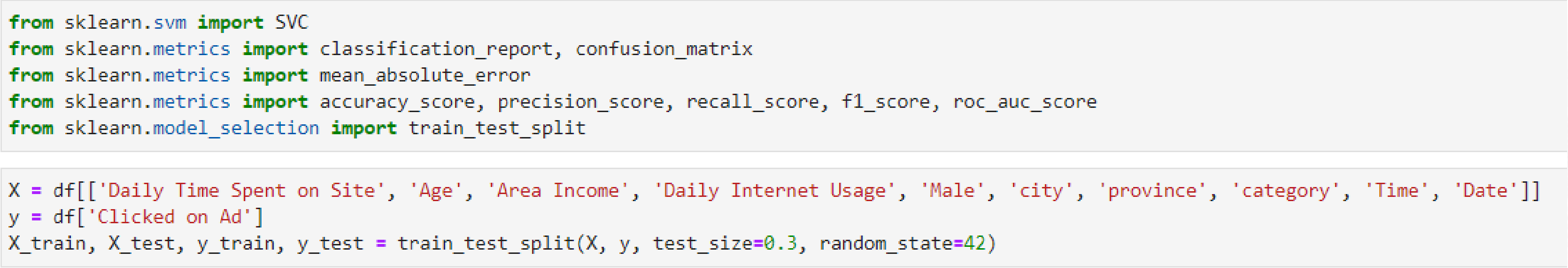
```
df['Male'].replace(['Perempuan', 'Laki-Laki'], [0, 1], inplace=True)
```

```
for i in ['city', 'province', 'category']:
    onehots = pd.get_dummies(df[i], prefix=i)
    df = df.join(onehots)
```

For more details, you can see Jupiter Notebook here

Modelling

- **Split Train Test**

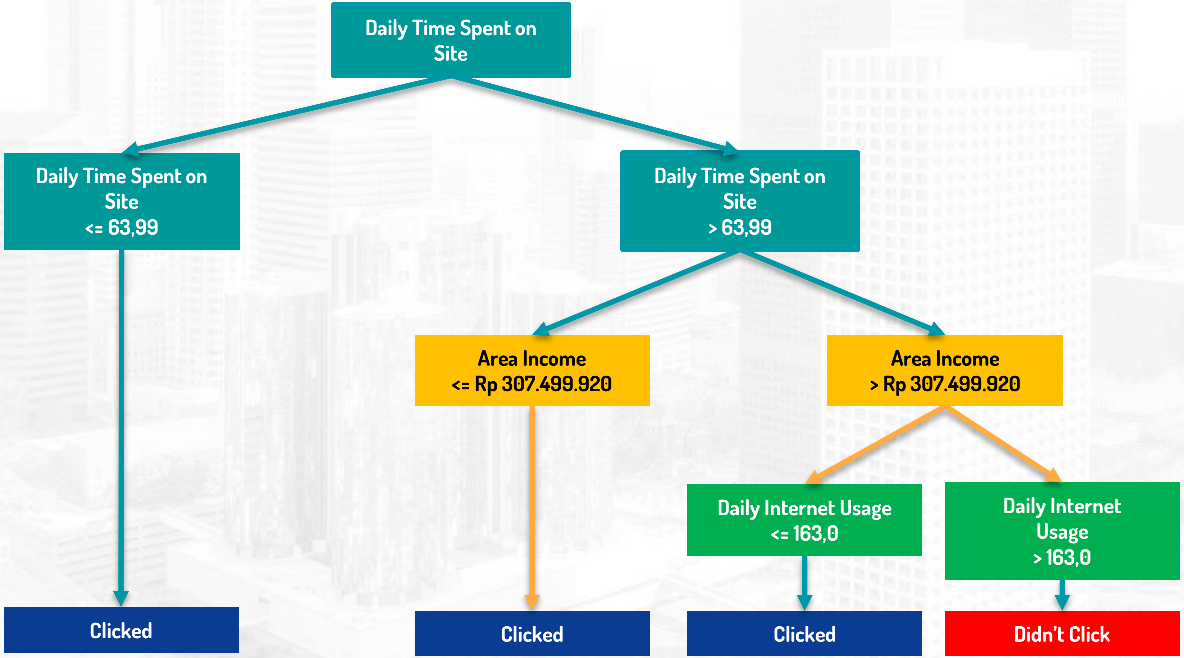


```
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score
from sklearn.model_selection import train_test_split
```

```
X = df[['Daily Time Spent on Site', 'Age', 'Area Income', 'Daily Internet Usage', 'Male', 'city', 'province', 'category', 'Time', 'Date']]
y = df['Clicked on Ad']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

For more details, you can see Jupiter Notebook here

Decision Tree Classifier


Daily Time Spent on Site
Daily Time Spent on Site <= 63,99
Daily Time Spent on Site > 63,99
Area Income <= Rp 307.499.920
Area Income > Rp 307.499.920
Daily Internet Usage <= 163,0
Daily Internet Usage > 163,0
Clicked
Clicked
Clicked
Didn't Click

Segmentation:

Visitors who have the potential to click on ads are:

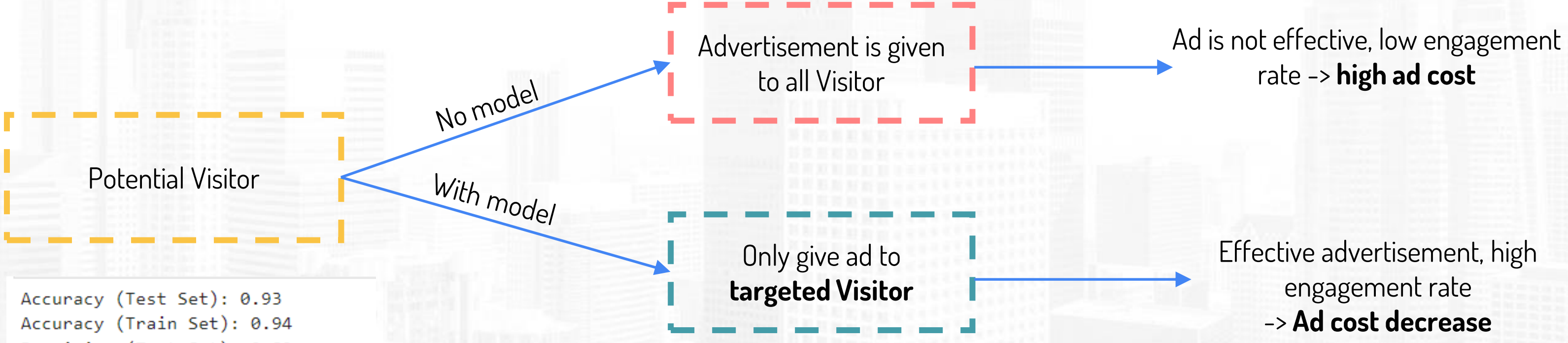
1. Visitors who have `**Daily Time Spent on Site**` <= 63.99
2. Visitor with `**Daily Time Spent on Site**` > 63.99, with `**Area Income**` > Rp 307,499,920, and `**Daily Internet Usage**` <= 163.0
3. Visitor with `**Daily Time Spent on Site**` > 63.99; `**Income Area**` <= IDR 307,499,920

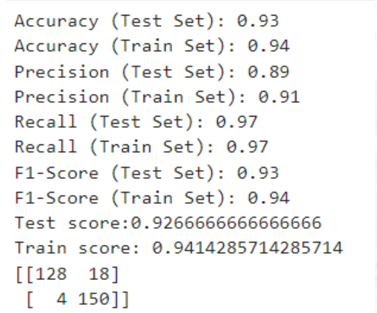
Visitors who have no potential to click on ads are:

1. Visitor with `**Daily Time Spent on Site**` > 63.99, with `**Area Income**` > Rp 307,499,920, and `**Daily Internet Usage**` > 163.0

For more details, you can see Jupiter Notebook here

Business Recommendation


Potential Visitor
No model
With model
Advertisement is given to all Visitor
Only give ad to **targeted Visitor**
Ad is not effective, low engagement rate -> **high ad cost**
Effective advertisement, high engagement rate -> **Ad cost decrease**



```
Accuracy (Test Set): 0.93
Accuracy (Train Set): 0.94
Precision (Test Set): 0.89
Precision (Train Set): 0.91
Recall (Test Set): 0.97
Recall (Train Set): 0.97
F1-Score (Test Set): 0.93
F1-Score (Train Set): 0.94
Test score:0.9266666666666666
Train score: 0.9414285714285714
[[128  18]
 [  4 150]]
```

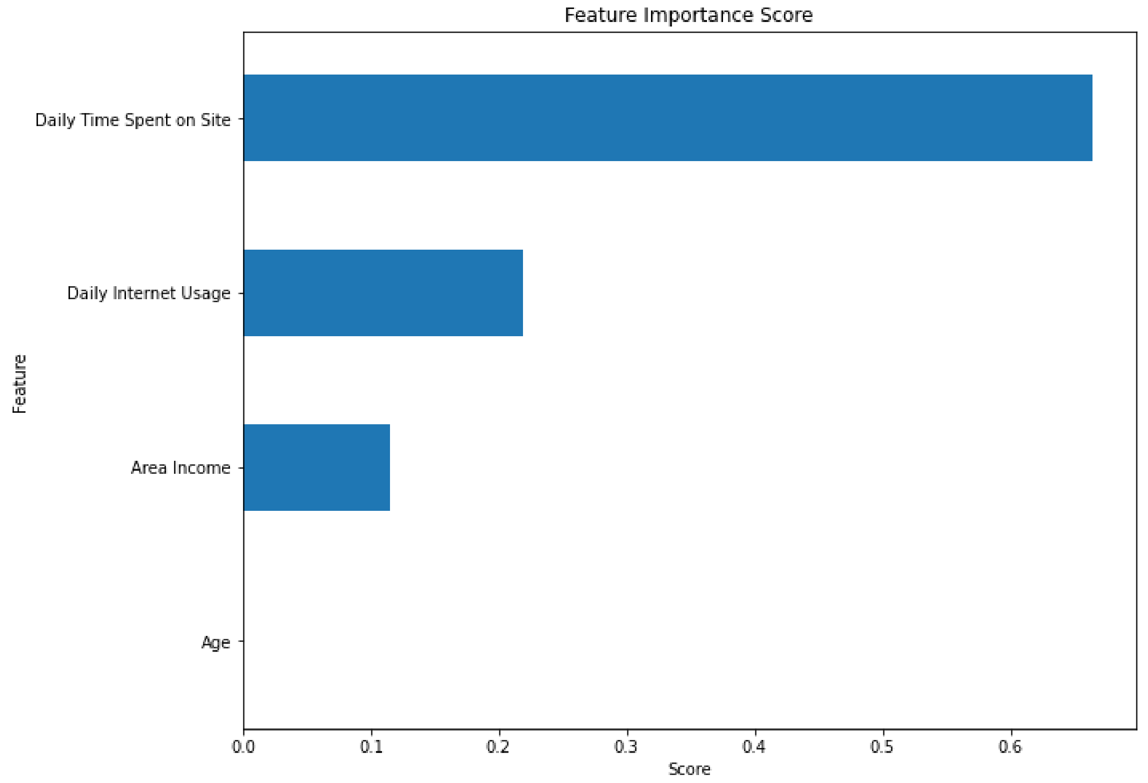
Evaluation :

Recall (Train Set) : 0,97
Recall (Test Set) : 0,97

	Engagement Rate		**Advertisement Cost** **Assume : (10k/Impression)**	
	Without ML	With ML	Without ML	With ML
Total Impression	1000	515	10000000	5150000
Total Engagement	500	500	48.50%	
Engagement Rate	50.00%	97.09%	Decrease in advertisement cost	
47% increase in engagement rate				

For more details, you can see Jupiter Notebook here

Business Recommendation



Business Recommendations:

- Ads should be placed on the Landing Page or Home Page, because it appears that customers do not spend too much time on the site.
- Ads are made more attractive and eye catchy, so that visitors are aware of ads
- Provide ads that match the categories that customers like