Predict Clicked Ads Customer Classification by using Machine Learning

Created by:
St S Bintang Pratama Dumatubun
nivandumatubun30@gmail.com
Nivan Dumatubun

Data Enthusiast. I have an interest in data and technology, i'm also pursuing experience and knowledge in technology through bootcamp and projects.

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

Overview

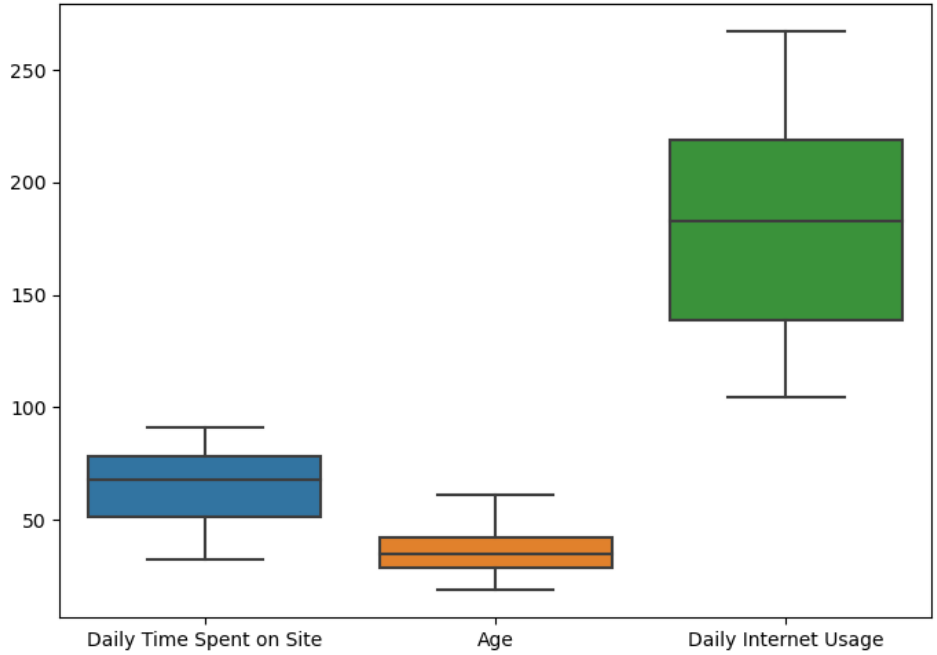
Sebuah perusahaan di Indonesia ingin mengetahui efektifitas sebuah iklan yang mereka tayangkan, hal ini penting bagi perusahaan agar dapat mengetahui seberapa besar ketercapainnya iklan yang dipasarkan sehingga dapat menarik customers untuk melihat iklan.

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.

Exploratory Data Analysis

Pada tahap *Exploratory Data Analysis*, dilakukan pembagian fitur menjadi numerical dan kategorikal. Selanjutnya dilakukan analisis terhadap kedua kategori tersebut

Univariate Analysis



Terlampir merupakan boxplot dari fitur numerical yang sudah ditentukan. Berdasarkan boxplot disamping, dapat dilihat distribusi dari masing-masing fitur.

1. ***Daily Time Spent on Site***

Boxplot menunjukkan bahwa sebagian besar pengguna menghabiskan waktu di situs dalam kisaran yang relatif seragam. Tidak ada pencilan yang mencolok di atas atau di bawah kuartil atas dan bawah.

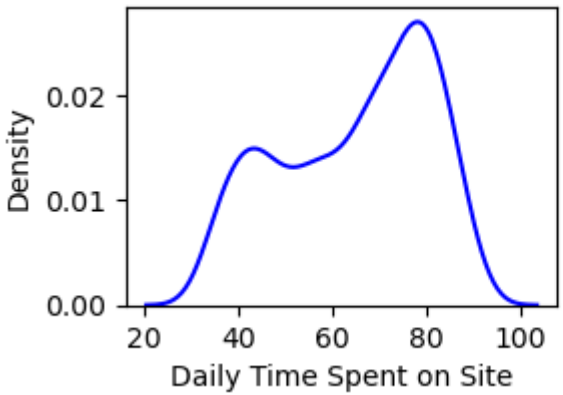
2. ***Age***

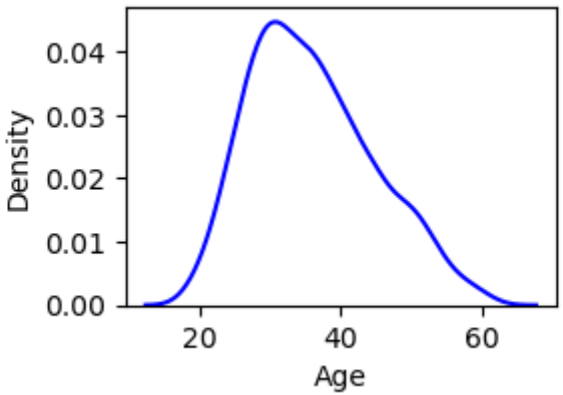
Boxplot menunjukkan bahwa distribusi umur cukup merata, dengan sedikit pencilan di atas kuartil atas.

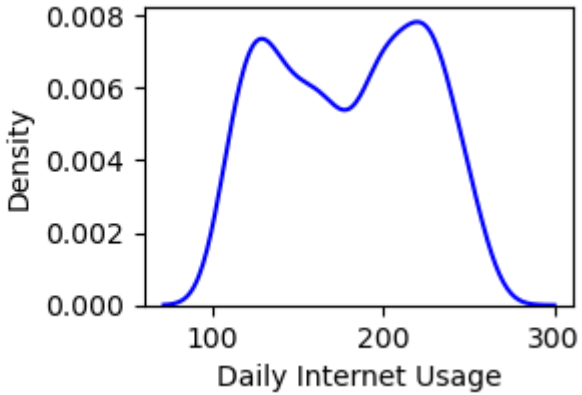
3. ***Daily Internet Usage***

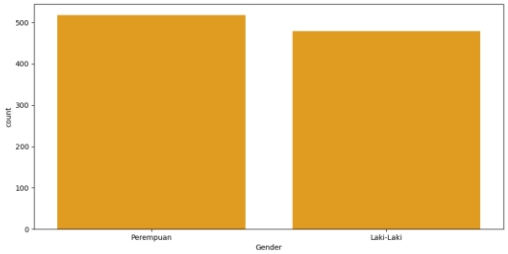
Boxplot menunjukkan bahwa sebagian besar pengguna menghabiskan internet harian dalam kisaran yang seragam, dengan sedikit pencilan di atas kuartil atas.

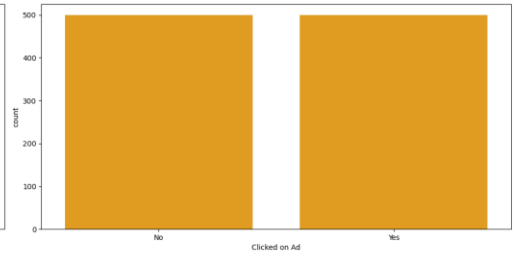
Exploratory Data Analysis

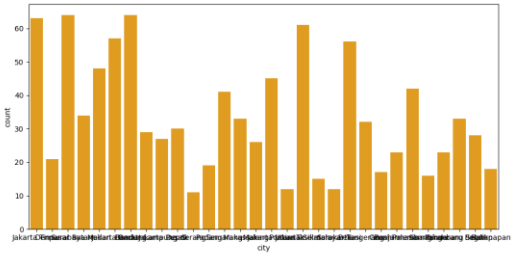


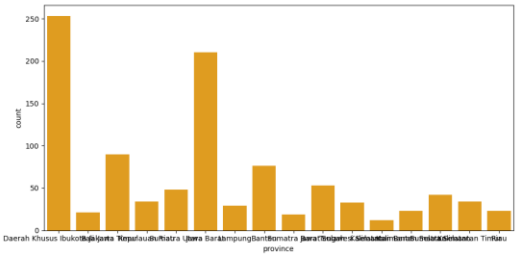


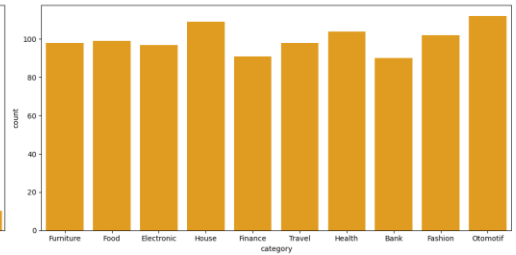






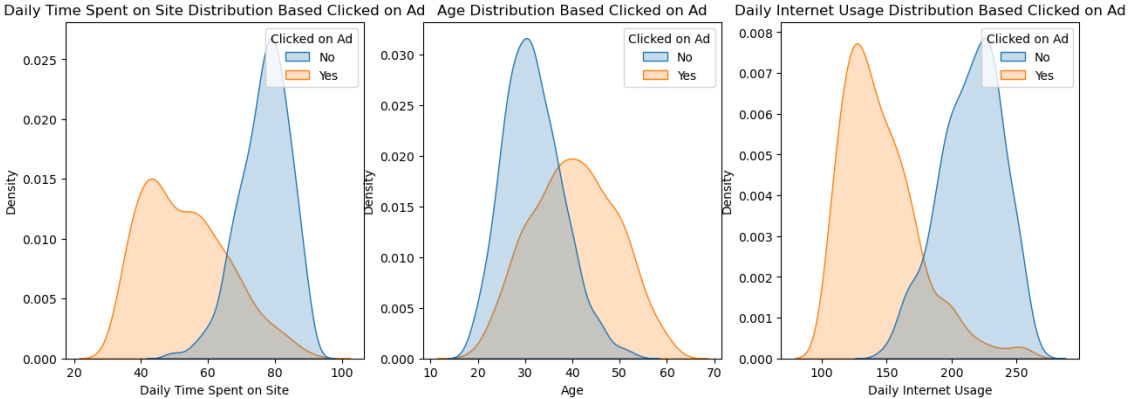






Exploratory Data Analysis

Bivariate Analysis



Pada *bivariate analysis* terdapat 3 grafik yaitu **Daily Time Spent on Site Distribution Based Clicked on Ad**, **Age Distribution Based Clicked on Ad**, dan **Daily Internet Usage Distribution Based Clicked on Ad**. Dari ketiga grafik tersebut masing-masing menunjukkan distribusi antara user yang melakukan **klik pada iklan** dan yang **tidak melakukan klik pada iklan**.

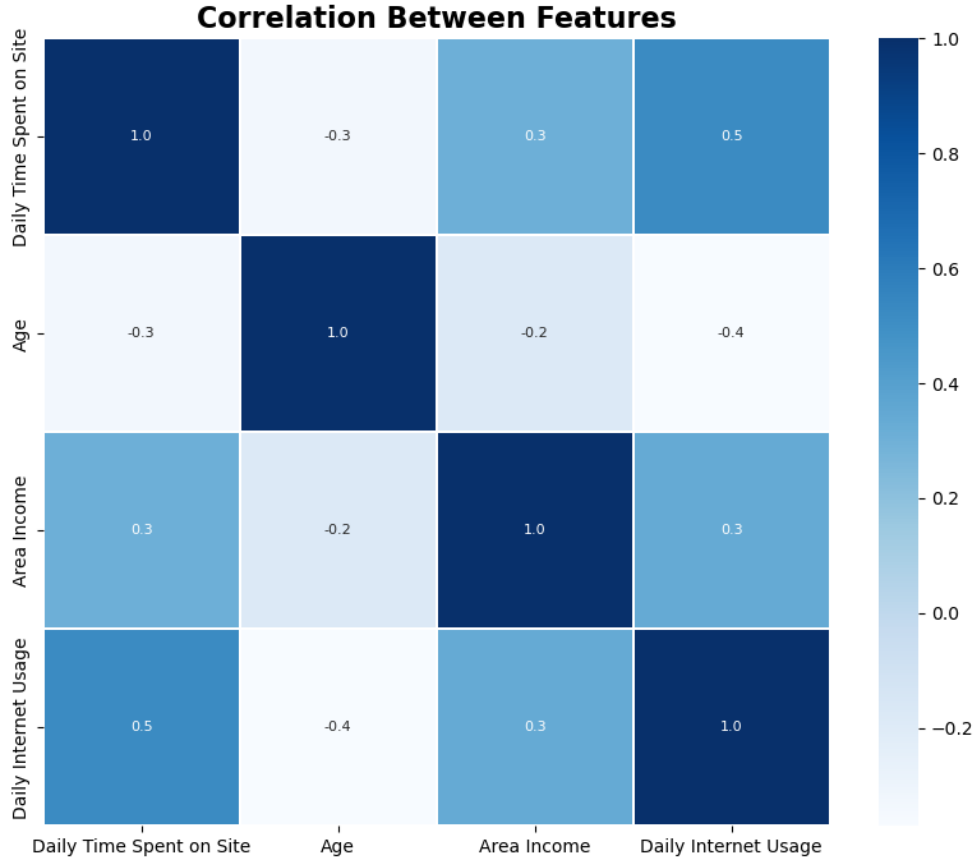
1. **Daily Internet Usage Distribution Based Clicked on Ad** :
Pada grafik hubungan ini, pengguna yang menghabiskan sedikit waktu (< 60 menit) untuk mengunjungi situs platform cenderung **melakukan klik pada iklan** sedangkan pengguna yang menghabiskan banyak waktu pada situs platform (> 60 menit) cukup banyak yang **tidak melakukan klik pada iklan**.

2. **Age Distribution Based Clicked on Ad** :
Pada grafik hubungan ini, pengguna yang berumur 20 hingga akhir 60 tahun cukup sering melakukan **klik pada iklan** namun tidak sebanyak pengguna yang berumur 15 hingga akhir 50 tahun yang **tidak melakukan klik pada iklan**.

3. **Daily Internet Usage Distribution Based Clicked on Ad** :
Pada grafik hubungan ini, kedua kategori cukup terpisah dengan baik. Pengguna dengan waktu penggunaan internet sedikit cenderung melakukan **klik pada iklan**, sedangkan untuk pengguna dengan waktu penggunaan internet yang cukup tinggi cenderung **tidak melakukan klik pada iklan**.

Exploratory Data Analysis

Multivariate Analysis



Multivariate Analysis dilakukan dengan memvisualisasikan heatmap disebelah kiri. Berdasarkan heatmap disamping, dapat dilihat bahwa hamper tidak ada korelasi antar fitur yang tinggi kecuali untuk korelasi antara **Daily Time Spent on Site** dan **Daily Internet Usage** yaitu sebesar **0.5**.

Data Cleaning and Preprocessing

Missing Value & Duplicated Rows

```
df5.isna().sum()
```

```
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
```

```
df5.duplicated().sum()
```

```
0
```

Pada pemeriksaan nilai kosong dan baris duplikat, terdapat nilai kosong pada fitur *Daily Time Spent on Site*, *Area Income*, *Daily Internet Usage*, dan *Gender*. Sedangkan tidak terdapat baris duplikat pada dataframe.

```
fill = ['Daily Time Spent on Site','Area Income','Daily Internet Usage']
for col in fill:
    med = df5[col].median()  # Hitung nilai rata-rata kolom
    df5[col].fillna(med, inplace=True)
```

```
df5['Gender'] = df5['Gender'].fillna(df5['Gender'].mode()[0])
```

Ini merupakan penanganan pada nilai kosong yang mana pada fitur numerikan diisi dengan median dan kolom kategorikal (*Gender*) diisi dengan modus. Sehingga tidak lagi terdapat nilai kosong pada semua fitur.

```
df5.isna().sum()
```

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

Data Cleaning and Preprocessing

Datetime Extraction

```
df5['Timestamp'] = pd.to_datetime(df5['Timestamp'])
```

Sebelum melakukan ekstraksi fitur, dilakukan terlebih dahulu perubahan tipe data dari object menjadi datetime. Perubahan tersebut dilakukan dengan potongan code seperti terlampir

```
df5['Weekday'] = df5['Timestamp'].dt.dayofweek

df5['Month'] = df5['Timestamp'].dt.month

df5['Hour'] = df5['Timestamp'].dt.hour
```

Selanjutnya melakukan ekstraksi kolom *Timestamp* menjadi beberapa bagian yang membentuk kolom baru. Dalam hal ini dibuat kolom baru berupa Bulan, Hari, dan Jam.

Data Cleaning and Preprocessing

Feature Encoding

Feature encoding dibagi menjadi 2 yaitu :

- Label Encoding : **Gender** dan **Clicked on Ad**
- One Hot Encoding : **Province** dan **Category**

```
map_gen = {'Laki-Laki' : 0, 'Perempuan' : 1}
df5['Gender'] = df5['Gender'].map(map_gen)
```

```
map_coa = {'Yes' : 1, 'No' : 0}
df5['Clicked on Ad'] = df5['Clicked on Ad'].map(map_coa)
```

Label encoding dilakukan pada fitur *Gender* dan *Clicked on Ad*. Pada *Gender*, 0 sebagai laki-laki dan 1 sebagai Perempuan. Pada *Clicked on Ad* 0 untuk No dan 1 untuk Yes.

```
df5['Pulau'] = np.where(((df5['province'] == 'Daerah Khusus Ibukota Jakarta') |
                         (df5['province'] == 'Jawa Barat') |
                         (df5['province'] == 'Jawa Tengah') |
                         (df5['province'] == 'Jawa Timur') |
                         (df5['province'] == 'Banten')), 'Jawa',
                        np.where(((df5['province'] == 'Riau') |
                                  (df5['province'] == 'Sumatra Utara') |
                                  (df5['province'] == 'Sumatra Barat') |
                                  (df5['province'] == 'Lampung') |
                                  (df5['province'] == 'Sumatra Selatan')), 'Sumatra',
                                 np.where(((df5['province'] == 'Kepulauan Riau') |
                                           (df5['province'] == 'Kalimantan Timur') |
                                           (df5['province'] == 'Kalimantan Selatan') |
                                           (df5['province'] == 'Kalimantan Barat')), 'Kalimantan',
                                          np.where(((df5['province'] == 'Bali')), 'Bali', 'Sulawesi')
                                         )
                                 )
                        )
```

Khusus untuk fitur province, dibagi kedalam beberapa pulau terlebih dahulu karena nilai unique nya sangat banyak, setelah dibagi menjadi beberapa pulau dilakukan feature encoding terhadap pulau-pulau tersebut.

```
df5 = pd.get_dummies(df5, columns=['Pulau'])
```

```
df5 = pd.get_dummies(df5, columns=['category'])
```

Train and Test Split

```
x = dfmod.drop(columns='Clicked on Ad').copy()
y = dfmod['Clicked on Ad'].copy()
```

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3,random_state=0)
```

```
print(x_train.shape)
print(x_test.shape)
```

```
(700, 23)
(300, 23)
```

Terlampir adalah potongan code untuk melakukan train test split data. Data dibagi menjadi x (fitur) dan y (target). Dengan melakukan import train_test_split dan data akan dibagi menjadi x_train, y_train, x_test, dan y_test. X_train memiliki dimensi 700 baris dan 23 kolom sedangkan x_test 300 baris dan 23 kolom.

Data Modelling

Experiment 1 (Without Normalization)

	model_name	model	accuracy	recall	precision
0	K-Nearest Neighbor	KNeighborsClassifier()	0.620000	0.595588	0.578571
1	Logistic Regression	LogisticRegression()	0.546667	0.000000	0.000000
2	Decision Tree	DecisionTreeClassifier()	0.906667	0.955882	0.855263
3	Random Forest	(DecisionTreeClassifier(max_features='sqrt', r...	0.946667	0.948529	0.934783
4	Gradient Boosting	([DecisionTreeRegressor(criterion='friedman_ms...	0.946667	0.955882	0.928571

Berikut adalah hasil modeling dengan menggunakan data default (preprocessing sederhana). Hasil dari modeling tersebut dapat dilihat bahwa **Random Forest Classifier** memiliki akurasi terbesar. Adapun metode lain yang cukup tinggi akurasinya adalah **Gradient Boosting**. Untuk beberapa model seperti logistic regression dan k-nearest neighbor akurasi yang dihasilkan tidak begitu bagus.

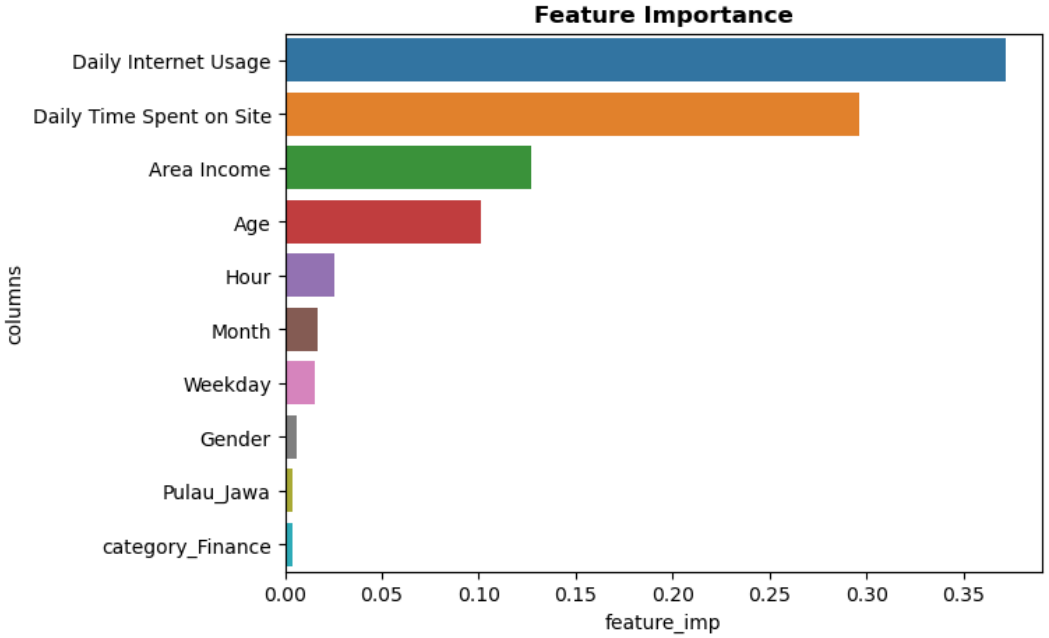
Data Modelling

Experiment 2 (With Normalization)

	model_name	model	accuracy	recall	precision
0	K-Nearest Neighbor	KNeighborsClassifier()	0.876667	0.897059	0.841379
1	Logistic Regression	LogisticRegression()	0.966667	0.955882	0.970149
2	Decision Tree	DecisionTreeClassifier()	0.916667	0.955882	0.872483
3	Random Forest	(DecisionTreeClassifier(max_features='sqrt', r...	0.963333	0.970588	0.949640
4	Gradient Boosting	([DecisionTreeRegressor(criterion='friedman_ms...	0.946667	0.955882	0.928571

Setelah menerapkan *min max scaler* diperoleh peningkatan signifikan pada beberapa model, terutama untuk model k-nearest neighbor dan logistic regression. Bahkan **logistic regression** menjadi model pertama yang paling tinggi akurasinya lalu disusul oleh **random forest**. Berdasarkan metode tersebut kita akan pilih **random forest** sebagai model terbaik karena memiliki akurasi yang paling tinggi. Logistic Regression juga bisa menjadi pilihan yang baik jika ada kendala tentang komputasi.

Feature Importance



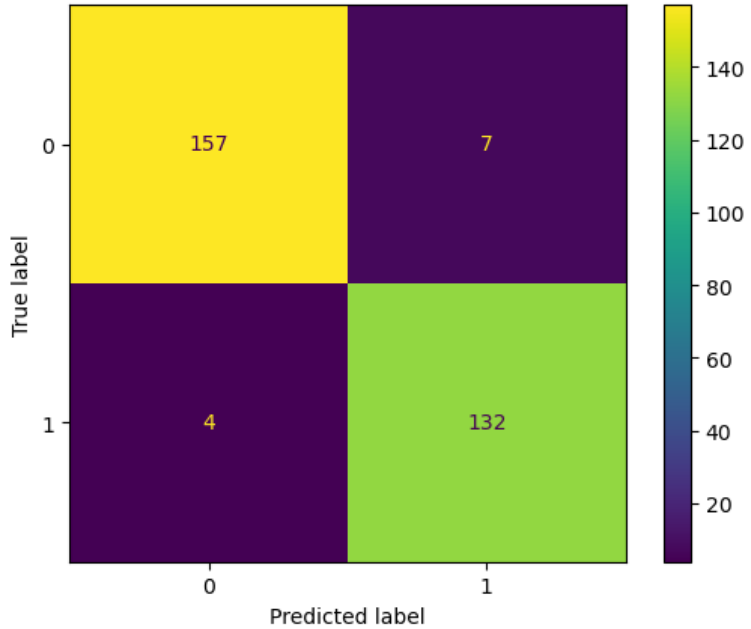
Dengan menggunakan model random forest kita mampu melihat feature yang paling penting dalam membangun model. Dalam hal ini, ditetapkan *threshold* untuk feature importance yaitu sebesar **0.10**, sehingga score importance **> 0.10** dianggap sebagai feature yang berpengaruh.

Berdasarkan metode random forest kita dapat melihat bahwa **daily internet usage** merupakan feature yang sangat penting dalam penentuan apakah user akan click atau tidak. Adapun feature penting lain adalah **daily time spent on site, income area, dan usia**.

Jika di combine insight yang kita peroleh dari proses EDA kita menjadi tahu bahwa ternyata penggunaan internet harian jika semakin tinggi maka peluang user akan click semakin kecil.

Model Evaluation

Confusion Matrix



Berdasarkan model random forest kita ingin melihat bagaimana performa model kita secara mendetail. Untuk itu kita akan menggunakan confusion matrix.

Confusion matrix yang dihasilkan random forest sangatlah baik. Kita dapat melihat kesalahan prediksi (**cell ungu**) berjumlah sangat sedikit (**bagian kanan atas dan kiri bawah**).

Dengan hasil berikut maka kita akan mendapatkan akurasi, precision, dan recall yang bagus.

Recommendations

Berdasarkan EDA, model prediksi, dan feature importance yang sudah dibuat dapat diberikan rekomendasi yaitu terdapat 4 fitur penting dalam memprediksi apakah customer akan melakukan klik pada iklan atau tidak. 4 Features importance tersebut antara lain :

1. **Daily Internet Usage**

Berdasarkan feature importance yang tinggi, fokuskan strategi pemasaran pada pengguna yang memiliki tingkat penggunaan internet harian tinggi. Hal ini dapat dilakukan dengan menyediakan konten iklan yang menarik dan relevan bagi pengguna yang aktif menggunakan internet setiap hari.

2. **Daily Time Spent on Site**

Fitur ini menunjukkan berapa lama pengguna menghabiskan waktu di situs web. Anda dapat memanfaatkan informasi ini dengan menyajikan iklan yang lebih menarik bagi pengguna yang menghabiskan waktu lebih lama di situs web Anda. Misalnya, dengan menampilkan iklan yang lebih interaktif atau menawarkan diskon khusus bagi pengguna yang aktif di situs Anda.

3. **Area Income**

Fitur ini menunjukkan tingkat pendapatan daerah tempat pengguna berada. Anda dapat mengadaptasi strategi pemasaran berdasarkan tingkat pendapatan ini. Misalnya, dengan menawarkan produk atau layanan yang sesuai dengan tingkat pendapatan mereka, atau dengan menyesuaikan harga atau penawaran khusus sesuai dengan tingkat pendapatan daerah.

4. **Age**

Usia pelanggan juga bisa menjadi faktor penting dalam menentukan kecenderungan mereka untuk mengklik iklan tertentu. Dalam kasus ini, Kalangan orang tua menjadi market yang potensial untuk market digital.

Untuk selengkapnya dapat dilihat disini

Simulation

Dengan asumsi :

- Untuk beriklan terhadap seorang user bisa menggunakan budget **10rb rupiah**
- Menggunakan data test sebagai alat simulasi sekitar 300 user dengan jumlah user pada masing-masing class sebanyak 164 dan 136 user.
- Setiap user yang convert kita akan mendapatkan keuntungan sebesar **12rb rupiah**

Simulasi :

1. Tanpa Machine Learning Model

- Kita akan menggunakan budget sekitar 300 * 10rb = 3jt rupiah untuk melakukan advertisement
- **Cost = 3jt**
- Sedangkan conversion rate yang akan kita dapatkan sebanyak 50%
- Karena hanya ada 136 yang convert maka kita akan mendapatkan 136 * 12rb = 1.63jt
- **Revenue = 1.63jt**
- **Profit = 1.63-3 = -1.37jt**

Berdasarkan simulasi di atas jika kita tidak menggunakan machine learning model dan maka kita akan mendapatkan ***potential loss*** **sebesar 1.37jt rupiah.**

Simulasi :

2. Dengan Menggunakan ML Model

- Kita akan melakukan advertisement hanya pada user yang berpotensi clicked (yang kita prediksi 1)
- Kita akan menggunakan budget sekitar 139 * 10rb = 1.39jt rupiah untuk melakukan advertisement
- **Cost = 1.39jt**
- Sedangkan conversion rate yang akan kita dapatkan sebanyak 132/139 = **94.96%**
- Dari 139 yang kita prediksi akan ada 132 user yang convert
- Maka kita akan mendapatkan 132 * 12rb = 1.58jt
- **Revenue = 1.58jt**
- **Profit = 1.58-1.39 = 190rb**

Berdasarkan simulasi di atas jika kita tidak menggunakan machine learning model dan maka kita akan mendapatkan **potential revenue sebesar 190rb rupiah.**

Untuk selengkapnya dapat dilihat disini

TERIMA KASIH!