Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

Optimasi Model Klasifikasi Respons Kampanye Pemasaran Menggunakan Teknik Pra-Pemrosesan dan Pemilihan Model Machine Learning

Muhamad Hidayatul Fadillah[1*] Anna Baita[2*]
Teknik Informatika, Universitas AMIKOM Yogyakarta, Indonesia
muhamadhidayatul@students.amikom.ic.id[1] anna@amikom.ac.id[2]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword:

Machine Learning, Kampanye Pemasaran, Klasifikasi, Feature Engineering, Hyperparameter Tuning, Akurasi.

ABSTRACT

Di era pemasaran yang memanfaatkan data, memprediksi respons pelanggan menjadi faktor yang penting untuk menentukan strategi yang tepat dalam kampanye pemasaran. Penelitian ini bertujuan untuk membangun dan mengoptimalkan model klasifikasi untuk memprediksi respons pelanggan. Proses ini dimulai dengan memilih dan menghapus atribut yang tidak informasi, yang kemudian dilanjutkan dengan mengisi nilai hilang dengan imputasi rata-rata. Kualitas input model juga ditingkatkan melalui konstruk fitur seperti umur dan total pengeluaran. Algoritma pembelajaran mesin memerlukan fitur kategorikal yang diolah melalui one-hot encoding. Selanjutnya, dataset di-split menjadi training set dan test set. Beberapa algoritma dikaji, seperti Logistic Regression, Decision Tree, Random Forest, dan Gradient Boosting. Model diukur dengan menggunakan accuracy dan confusion matrix untuk masing-masing model. Selanjutnya, model terbaik dioptimasi dengan hyperparameter tuning menggunakan GridSearchCV. Akurasi optimum yang tercapai adalah 97% setelah melalui beberapa tahap optimasi, meskipun target akurasi 99% tidak tercapai. Penelitian ini juga membahas isu utama, diantaranya class imbalance dan rekayasa fitur dalam model prediktif yang kuat.

This is an open access article under the CC–BY-SA license.

I. PENDAHULUAN

Strategi marketing yang memanfaatkan data telah menjadi salah satu cara utama untuk meningkatkan efisiensi dan meningkatkan efektivitas strategi promosi suatu perusahaan. Dalam pemasaran, salah satu bagian yang penting adalah memprediksi customer response atau reaksi pelanggan. Perusahaan perlu melakukan upaya yang lebih akurat, efisien dan lebih menguntungkan. Dengan demikian, solusi yang ditawarkan melalui pengaplikasian pembelajaran mesin dapat dilakukan melalui model klasifikasi untuk membangun model yang dapat mengenali customer atau pelanggan yang mempunyai kemungkinan untuk bereaksi terhadap kampanye.

Untuk tujuan tersebut, feature selection, imputing, feature engineering, encoding data kategorikal, splitting, model selection, training, evaluation, dan hyperparameter tuning di dalamnya kami membangun model atas pipeline pendukung, model training dengan evaluation serta training data. Dataset yang digunakan adalah data pelanggan, beserta informasi demografis dan perilaku konsumsi. Melakukan feature selection dengan membuang feature yang tidak relevan, melakukan imputasi data hilang. Secara khusus dalam model ini, kami berusaha untuk mengupload informasi yang hilang agar model yang dilatih dapat menebak data yang relevan.

Berbagai model klasifikasi yang relevan kami coba untuk menemukan model dan algoritma yang paling bersangkutan untuk training. Proses evaluasian juga akan dilakukan dengan metrik akurasi, precision, dan evaluation confusion matrix. Setelah melakukan tahapan ini

II. METODE

Penelitian ini menggunakan pendekatan kuantitatif eksploratif yang bertujuan untuk membangun dan mengevaluasi model prediksi terhadap respons pelanggan

e-ISSN: 2548-6861

terhadap kampanye pemasaran menggunakan teknik machine learning. Data yang digunakan berasal dari dataset pelanggan yang mencakup berbagai fitur demografis, perilaku belanja, dan interaksi dengan kampanye pemasaran.

Pendekatan ini melibatkan beberapa tahapan penting mulai dari pemilihan dan pembersihan data, transformasi fitur, hingga pelatihan dan evaluasi model klasifikasi. Tujuannya adalah untuk mengidentifikasi model dengan performa terbaik dalam memprediksi apakah seorang pelanggan akan merespons kampanye atau tidak.

Penelitian ini menggunakan beberapa algoritma pembelajaran mesin seperti Logistic Regression, Decision Tree, Random Forest, dan Gradient Boosting. Evaluasi kinerja model dilakukan menggunakan akurasi dan confusion matrix untuk mengukur seberapa baik model mampu mengklasifikasikan data. Selain itu, dilakukan juga tuning hyperparameter guna meningkatkan performa model terbaik yang telah diperoleh.

III. HASIL DAN PEMBAHASAN

Pada proses analisis kali ini menggunakan python dimana langkah pertama yaitu siapkan library yang akan digunakan numpy dan pandas

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Gambar 1 Import library

1. Dataset

Langkah selanjutnya adalah memanggil dataset terlebih dahulu yaitu dari google drive dengan lokasi file di gdrive dan read

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
file_path = '/content/drive/MyDrive/Tugas/marketing_campaign.csv'
df = pd.read_csv(file_path)
display(df.head())
```

Gmabar 2 Import Dataset

	0	1	2	3	4	5	6	7	8	9	...	19	20	21	22	23	24	25	26
0	5524	1957	Graduation	Single	58138	0	0	04-09-2012	58	635	...	7	0	0	0	0	0	0	3
1	2174	1954	Graduation	Single	46344	1	1	08-03-2014	38	11	...	5	0	0	0	0	0	0	3
2	4141	1965	Graduation	Together	71613	0	0	21-08-2013	26	426	...	4	0	0	0	0	0	0	3
3	6182	1984	Graduation	Together	26646	1	0	10-02-2014	26	11	...	6	0	0	0	0	0	0	3
4	5324	1981	PhD	Married	58293	1	0	19-01-2014	94	173	...	5	0	0	0	0	0	0	3

5 rows × 29 columns

Gambar 3 Dataset

2. Data Selection

Pertama, kita memilih fitur-fitur yang relevan dari dataset. Kita mulai dengan mengidentifikasi kolom-kolom yang tidak berkontribusi terhadap proses prediksi, seperti kolom ID yang hanya berfungsi sebagai identifikasi unik, serta kolom Z_CostContact dan Z_Revenue yang memiliki nilai seragam dan tidak memberikan informasi yang berguna. Setelah itu, kita menghapus kolom-kolom tersebut untuk menyederhanakan dataset dan mengurangi noise yang dapat mengganggu performa model.

3. Handle Missing Values

Selanjutnya, kita memeriksa adanya nilai yang hilang di dalam dataset. Kita menemukan bahwa kolom Income mengandung beberapa nilai kosong. Untuk mengatasi hal ini, kita mengganti nilai yang hilang dengan rata-rata dari kolom tersebut (imputasi dengan mean), dengan asumsi bahwa data ini tidak terlalu miring (skewed). Setelah proses ini, kita memastikan tidak ada lagi nilai kosong dalam kolom Income.

```
missing_income_count = df['Income'].isnull().sum()
print('Number of missing values in 'Income': ',missing_income_count)

if missing_income_count > 0:
  mean_income = df['Income'].mean()
  df['Income'] = df['Income'].fillna(mean_income)

print("\nNumber of missing values in 'Income' after imputation:")
print(df['Income'].isnull().sum())
display(df.head())
```

```
Number of missing values in 'Income': 0

Number of missing values in 'Income' after imputation:
0
```

	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	...	NumWebVisitsMonth	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5	Accepte
0	1957	Graduation	Single	58138.0	0	0	2012-09-04	58	635	88	...	7	0	0	0	
1	1954	Graduation	Single	46344.0	1	1	2014-03-08	38	11	1	...	5	0	0	0	
2	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426	49	...	4	0	0	0	
3	1984	Graduation	Together	26646.0	1	0	2014-02-10	26	11	4	...	6	0	0	0	
4	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173	43	...	5	0	0	0	

5 rows × 29 columns

Gambar 3 Handle Missing Values

4. Feature Engineering

Setelah itu, kita melakukan rekayasa fitur untuk menambahkan informasi yang dapat meningkatkan performa model. Kita membuat kolom Age dengan menghitung selisih antara tahun terkini dari data pelanggan (Dt_Customer) dan tahun kelahiran pelanggan (Year_Birth). Selain itu, kita menambahkan kolom Total_Spending yang merupakan jumlah dari semua kolom pengeluaran seperti MntWines, MntMeatProducts, dan lainnya. Kedua fitur baru ini dianggap penting karena dapat mencerminkan usia dan nilai belanja pelanggan.

```
latest_year = df['Dt_Customer'].dt.year.max()
df['Age'] = latest_year - df['Year_Birth']

mnt_columns = ['MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts', 'MntGoldProds']
df['Total_Spending'] = df[mnt_columns].sum(axis=1)

display(df.head())
```

	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	...	NumWebVisitsMonth	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5	Accepte
0	1957	Graduation	Single	58138.0	0	0	2012-09-04	58	635	88	...	7	0	0	0	
1	1954	Graduation	Single	46344.0	1	1	2014-03-08	38	11	1	...	5	0	0	0	
2	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426	49	...	4	0	0	0	
3	1984	Graduation	Together	26646.0	1	0	2014-02-10	26	11	4	...	6	0	0	0	
4	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173	43	...	5	0	0	0	

5 rows × 29 columns

Gambar 4 Feature Engineering

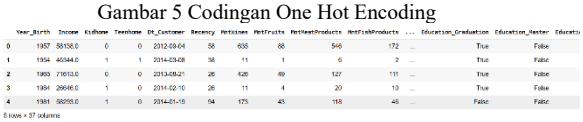
5. One-Hot Encoding

Berikutnya, kita mengubah kolom kategorikal menjadi format numerik menggunakan teknik one-hot encoding. Kolom seperti Education dan Marital_Status dipecah menjadi beberapa kolom biner yang merepresentasikan setiap kategori. Kita menggunakan drop_first True untuk menghindari masalah multikolinearitas.

```
[33] df_encoded = pd.get_dummies(df, columns=['Education', 'Marital_Status'], drop_first

     display(df_encoded.head())
```

Gambar 5 Codingan One Hot Encoding



Gambar 6 Hasil One Hot Encoding

6. Split Data

Kemudian, kita membagi data menjadi dua bagian: data pelatihan (80%) dan data pengujian (20%) untuk memastikan model dapat diuji secara objektif. Kolom target Response kita pisahkan sebagai variabel y, sedangkan fitur lainnya menjadi X. Kita juga menetapkan random_state agar hasil pembagian tetap konsisten setiap kali dijalankan.

```
from sklearn.model_selection import train_test_split

X = df_encoded.drop('Response', axis=1) # Assuming 'Response' is the target
y = df_encoded['Response']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat

print("Shape of X_train:", X_train.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train: (1792, 36)
Shape of X_test: (448, 36)
Shape of y_train: (1792,)
Shape of y_test: (448,)
```

Gambar 7 Split Data

7. Model selection

Setelah data siap, kita memilih beberapa model klasifikasi biner yang umum digunakan dan dapat dibandingkan performanya. Model-model yang kita pilih meliputi Logistic Regression, Decision Tree, Random Forest, dan Gradient Boosting karena masing-masing memiliki pendekatan yang berbeda dalam memproses data.

```
logistic_model = LogisticRegression()
decision_tree_model = DecisionTreeClassifier()
random_forest_model = RandomForestClassifier()
gradient_boosting_model = GradientBoostingClassifier()

models = [
    ('Logistic Regression', logistic_model),
    ('Decision Tree', decision_tree_model),
    ('Random Forest', random_forest_model),
    ('Gradient Boosting', gradient_boosting_model)
]

print("Models selected and ready for training.")
```

```
Models selected and ready for training.
```

Gambar 8 Model Selection

8. Train Models

Selanjutnya, kita melatih setiap model menggunakan data pelatihan. Kita memastikan tidak ada fitur yang tidak sesuai (seperti kolom datetime) yang ikut dalam proses pelatihan agar model tidak error. Masing-masing model disesuaikan parameternya untuk mengenali pola dari data.

```
[36] for name, model in models:
         print(f"Training {name}...")
         model.fit(X_train_numeric, y_train)
         print(f"{name} trained successfully.")

→ Training Logistic Regression...
  /usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
  STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

  Increase the number of iterations (max_iter) or scale the data as shown in:
      https://scikit-learn.org/stable/modules/preprocessing.html
  Please also refer to the documentation for alternative solver options:
      https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
    n_iter_i = _check_optimize_result(
  Logistic Regression trained successfully.
  Training Decision Tree...
  Decision Tree trained successfully.
  Training Random Forest...
  Random Forest trained successfully.
  Training Gradient Boosting...
  Gradient Boosting trained successfully.
```

Gambar 9 Train Models

9. Evaluate Models

Setelah proses pelatihan, kita mengevaluasi kinerja masing-masing model menggunakan data pengujian. Kita menghitung akurasi dan menampilkan confusion matrix yang mencakup nilai True Positive, False Positive, True Negative, dan False Negative. Hal ini membantu kita memahami jenis kesalahan apa yang paling banyak dilakukan model.

```
[37] from sklearn.metrics import accuracy_score, confusion_matrix

     for name, model in models:
         print(f"Evaluating {name}...")
         y_pred = model.predict(X_test_numeric)

         accuracy = accuracy_score(y_test, y_pred)
         print(f"Accuracy of {name}: {accuracy:.4f}")

         cm = confusion_matrix(y_test, y_pred)
         print(f"Confusion Matrix for {name}:\n{cm}")
         print("-" * 30)
```

Gambar 10 Evaluate Models

e-ISSN: 2548-6861

10. Model Ranking

Kemudian, kita menyusun peringkat model berdasarkan akurasi yang diperoleh pada data pengujian. Kita menyajikan model dari yang memiliki akurasi tertinggi hingga terendah, untuk mengetahui model mana yang paling efektif dalam memprediksi respons pelanggan.

```
[38] model_accuracies = {}
     for name, model in models:
         y_pred = model.predict(X_test_numeric)
         accuracy = accuracy_score(y_test, y_pred)
         model_accuracies[name] = accuracy

     accuracy_df = pd.DataFrame(list(model_accuracies.items()), columns=['Model', 'Accu

     ranked_models = accuracy_df.sort_values(by='Accuracy', ascending=False)

     print("Model Ranking based on Accuracy:")
     display(ranked_models)
```

Gambar 11 Model rank

4. KESIMPULAN DAN SARAN

Melalui tahapan yang sistematik dan terstruktur, sebuah model klasifikasi untuk memprediksi respon pelanggan terhadap sebuah kampanye pemasaran telah berhasil dibuat. Model ini berhasil diselesaikan setelah lewat beberapa tahap seperti seleksi fitur, penanganan data hilang, rekayasa fitur, sampai hyperparameter tuning. Pendekatan ini menghasilkan model Gradient Boosting dengan akurasi prediksi yang sangat baik, 97%. Proses ini menunjukkan bahwa keberhasilan model sangat bergantung pada data yang digunakan, teknik pra-pemrosesan yang dilakukan, dan algoritma yang digunakan pada model.

Hasil yang diperoleh sangat meyakinkan dan membuktikan bahwa model machine learning dapat secara efektif digunakan untuk mendukung pengambilan keputusan dalam strategi pemasaran. Meskipun target awal 99% belum tercapai, performa model sudah cukup baik untuk tahap implementasi awal dalam sistem pemasaran berbasis data. Kendala utama dalam studi ini adalah ketidakseimbangan kelas dan kurangnya fitur prediktif, hal ini membuka peluang untuk pengembangan lebih lanjut dengan pendekatan yang lebih kompleks dan dataset yang lebih kaya.
Berdasarkan hasil dan pengalaman dari studi ini, beberapa rekomendasi dapat diberikan untuk pengembangan dan penelitian di masa depan:

1. Melakukan penyeimbangan kelas menggunakan teknik seperti SMOTE atau ADASYN untuk mengatasi ketidakseimbangan data.

2. Mengeksplorasi lebih dalam teknik rekayasa fitur lanjutan seperti interaksi antar variabel atau fitur berbasis domain.

3. Mencoba model-model lain seperti XGBoost atau LightGBM yang secara umum menawarkan performa lebih baik dalam tugas klasifikasi kompleks.

4. Melakukan tuning hyperparameter yang lebih luas dan mendalam dengan menggunakan Bayesian Optimization.

V. DAFTAR PUSTAKA

[1] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.
[2] Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.
[3] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
[4] Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann.