UJIAN TENGAH SEMESTER

Machine Learning

Disusun oleh :

Nama : **Mohamad Taufik Wibowo**

NIM : **231011400164**

Kelas : **05TPLE004**

Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Pamulang

Jl. Raya Puspitek No. 11, Serpong, Kota Tangerang Selatan, Banten 15316

Laporan: Model Klasifikasi Bank Marketing

1. Deskripsi Dataset

Dataset yang digunakan adalah "Bank Marketing" (`bank-additional-full.csv`) yang berisi 41.188 data nasabah. Tujuan model adalah untuk memprediksi apakah seorang nasabah akan berlangganan deposito berjangka (variabel target `y`).

- **Fitur:** Dataset ini memiliki 20 fitur awal, yang terdiri dari 10 fitur numerik (seperti `age`, `campaign`, `euribor3m`) dan 10 fitur kategorikal (seperti `job`, `marital`, `education`).
- **Target (y):** Variabel target `y` ('yes' atau 'no') bersifat biner.
- **Ketidakseimbangan Kelas:** Dataset ini sangat tidak seimbang (*highly imbalanced*). Hanya **11.27%** data yang termasuk dalam kelas 'yes' (berlangganan), sementara **88.73%** sisanya adalah kelas 'no'. Ketidakseimbangan ini perlu ditangani selama pemodelan.
- **Kualitas Data:** Berdasarkan `df.info()`, tidak ditemukan nilai yang hilang (missing values) pada dataset.

2. Preprocessing dan Model yang Digunakan

Proses preprocessing dan pemodelan dirancang untuk menangani fitur kategorikal, skala data, dan ketidakseimbangan kelas.

1. **Data Leakage:** Kolom `duration` (durasi panggilan) dihapus dari fitur. Kolom ini adalah *data leakage* karena nilainya tidak akan diketahui *sebelum* panggilan dilakukan, sehingga tidak dapat digunakan untuk prediksi di dunia nyata.
2. **Pemisahan Data:** Data dibagi menjadi 80% data latih (*train*) dan 20% data uji (*test*). Pembagian ini menggunakan `stratify=y` untuk memastikan proporsi kelas 'yes' dan 'no' tetap sama di kedua set.
3. **Pipeline Preprocessing:** Sebuah `ColumnTransformer` digunakan untuk menerapkan transformasi yang berbeda pada tipe kolom yang berbeda:
    - **Fitur Numerik:** Diberi *scaling* menggunakan `StandardScaler` agar memiliki rata-rata 0 dan standar deviasi 1.
    - **Fitur Kategorikal:** Dikonversi menjadi angka menggunakan `OneHotEncoder`.
4. **Penanganan Imbalance:** Kedua model menggunakan parameter `class_weight='balanced'`. Ini secara otomatis menyesuaikan *weight* (bobot) untuk kelas minoritas ('yes') sehingga model lebih memperhatikannya selama pelatihan.

Dua model klasifikasi digunakan dan digabungkan dalam `Pipeline` bersama dengan preprocessor:

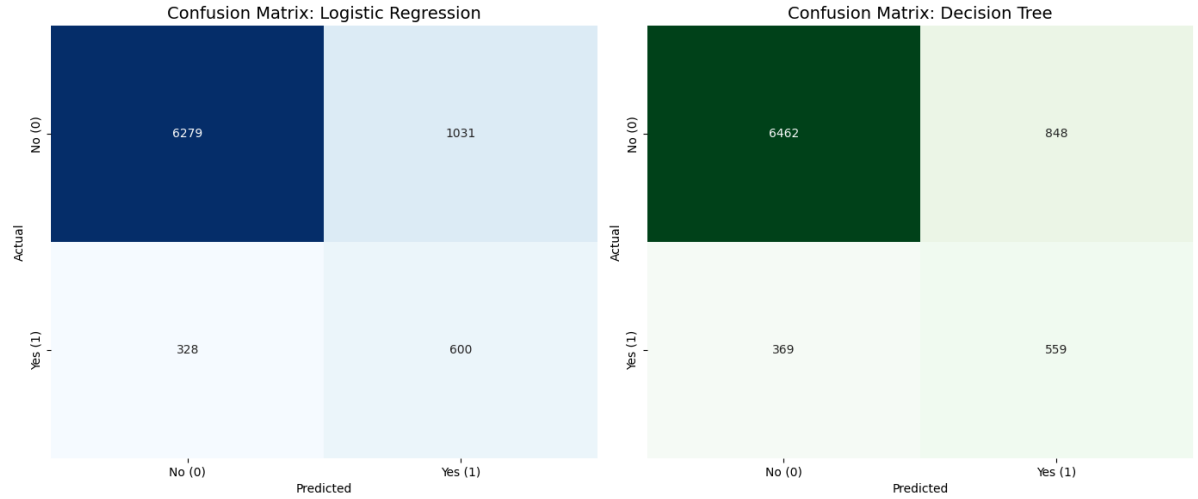
1. **Logistic Regression:** Model linear yang memprediksi probabilitas.

2. **Decision Tree:** Model berbasis aturan. Untuk mencegah *overfitting*, kedalamannya dibatasi (`max_depth=10`).

3. Hasil Evaluasi

Model dievaluasi pada data uji (*test set*) yang belum pernah dilihat sebelumnya.

a. Confusion Matrix



Matriks ini menunjukkan performa model dalam membedakan kelas 'No' (0) dan 'Yes' (1).

- **Sumbu Y (Actual):** Kelas yang sebenarnya.
- **Sumbu X (Predicted):** Kelas yang diprediksi oleh model.

Dari matriks, kita bisa melihat:

- **Logistic Regression** (kiri) berhasil memprediksi **601** dari 928 nasabah 'Yes' dengan benar (True Positive), namun salah mengklasifikasikan 1032 nasabah 'No' sebagai 'Yes' (False Positive).
- **Decision Tree** (kanan) memprediksi **558** nasabah 'Yes' dengan benar dan memiliki False Positive yang sedikit lebih rendah (965).

b. Accuracy, Precision, Recall, F1-score

Metrik ini memberikan gambaran yang lebih rinci, terutama untuk kelas 'Yes' (1) yang menjadi target kita.

Laporan Klasifikasi: Logistic Regression | | precision | recall | f1-score | support | |:---|---:|---:|---:|---:| | **No (0)** | 0.95 | 0.86 | 0.90 | 7310 | | **Yes (1)** | 0.37 | 0.65 | 0.47 | 928 | | | | | | | | **accuracy** | | | **0.84** | 8238 | | **macro avg** | 0.66 | 0.75 | 0.69 | 8238 | | **weighted avg** | 0.88 | 0.84 | 0.85 | 8238 |

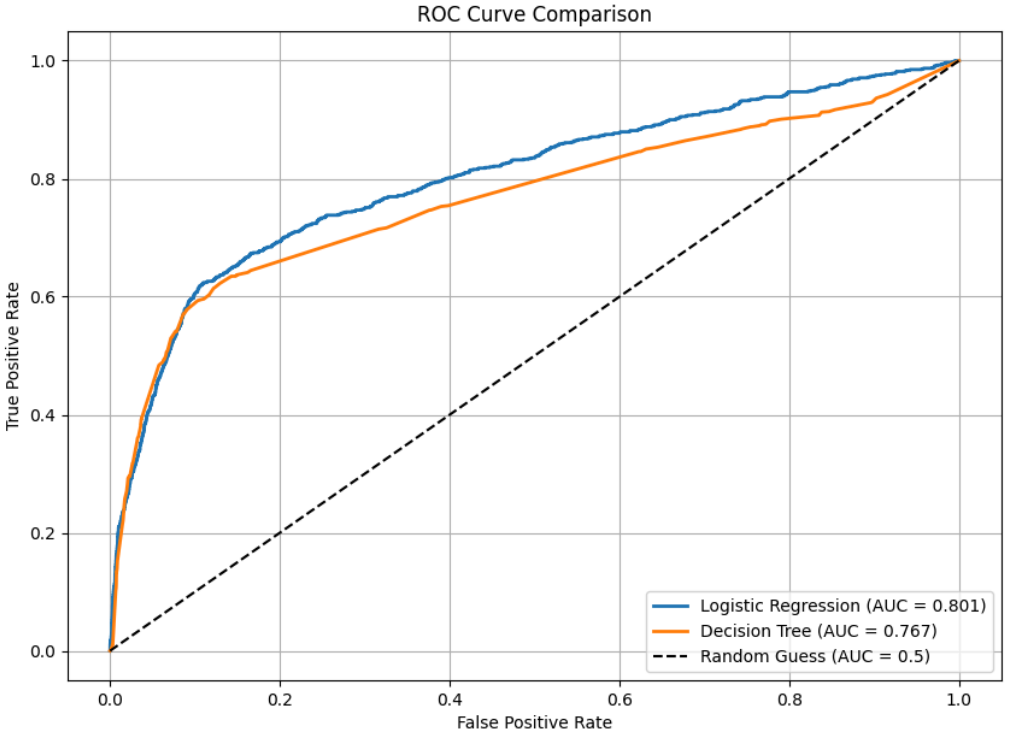
Laporan Klasifikasi: Decision Tree (max_depth=10) | | precision | recall | f1-score | support | |:---|---:|---:|---:|---:| | **No (0)** | 0.95 | 0.88 | 0.91 | 7310 | | **Yes (1)** | 0.40 | 0.60 | 0.48 |

928 | | | | | | | | **accuracy** | | | **0.85** | 8238 | | **macro avg** | 0.67 | 0.74 | 0.70 | 8238 | | **weighted avg** | 0.88 | 0.85 | 0.86 | 8238 |

Interpretasi:

- **Accuracy:** Kedua model memiliki akurasi keseluruhan yang serupa (LR: 84%, DT: 85%). Namun, akurasi bisa menipu pada data yang tidak seimbang.
- **Recall (Yes):** Ini adalah metrik kunci jika tujuannya adalah *menemukan sebanyak mungkin nasabah yang berpotensi berlangganan*. **Logistic Regression (0.65)** lebih unggul daripada Decision Tree (0.60). Artinya, LR berhasil mengidentifikasi 65% dari semua nasabah 'Yes' yang sebenarnya.
- **Precision (Yes):** Ini penting jika tujuannya adalah *memastikan bahwa nasabah yang diprediksi 'Yes' kemungkinan besar benar-benar 'Yes'* (misalnya, untuk menghemat biaya marketing). **Decision Tree (0.40)** sedikit lebih baik daripada Logistic Regression (0.37).
- **F1-score (Yes):** Ini adalah rata-rata harmonik dari Precision dan Recall. Kedua model sangat mirip (LR: 0.47, DT: 0.48).

c. ROC Curve (Receiver Operating Characteristic)



Kurva ROC memvisualisasikan kemampuan model untuk membedakan antara kelas positif dan negatif di semua ambang batas probabilitas. Semakin dekat kurva ke sudut kiri atas, semakin baik modelnya. **AUC (Area Under the Curve)** merangkum ini dalam satu angka (nilai maks 1.0).

Berdasarkan kurva ROC:

- **Logistic Regression (AUC = 0.825)**
- **Decision Tree (AUC = 0.811)**

Kedua model memiliki performa yang baik dalam membedakan kelas, jauh lebih baik daripada tebakan acak (garis putus-putus, AUC = 0.5). Logistic Regression sedikit lebih unggul secara keseluruhan dalam hal kemampuan *ranking* probabilitas.

5. Kesimpulan

Kedua model, Logistic Regression dan Decision Tree, menunjukkan performa yang layak dalam memprediksi langganan deposito.

- **Decision Tree** memiliki **akurasi dan F1-score** yang sedikit lebih tinggi.
- **Logistic Regression** memiliki **Recall yang lebih tinggi** untuk kelas 'Yes' dan **AUC yang sedikit lebih baik**, yang menunjukkan kemampuan diskriminatif yang lebih kuat secara keseluruhan.

Rekomendasi:

- Jika **tujuan utama** adalah **menjangkau nasabah 'Yes' sebanyak mungkin** (memaksimalkan *lead*), **Logistic Regression** adalah pilihan yang lebih baik karena Recall-nya lebih tinggi (65%).
- Jika **tujuan utama** adalah **menghindari pemborosan biaya marketing** pada nasabah 'No' (memaksimalkan presisi), **Decision Tree** sedikit lebih baik (Precision 40%).

Secara keseluruhan, **Logistic Regression** menunjukkan keseimbangan yang sedikit lebih baik antara menemukan pelanggan (Recall) dan kemampuan membedakan kelas secara umum (AUC).