Laporan Analisis dan Klasifikasi Data Diabetes

1. Pendahuluan

Dataset **diabetic_data.csv** yang digunakan dalam analisis ini berisi 101,766 entri dan 50 kolom yang terkait dengan rekam medis pasien diabetes. Tujuan utama dari analisis ini adalah untuk membangun model klasifikasi yang dapat memprediksi kemungkinan pasien akan **readmitted** (kembali dirawat) berdasarkan fitur medis dan demografi, seperti **waktu dirawat**, **obat-obatan yang digunakan**, dan **karakteristik demografi pasien**. Fitur targetnya adalah status **readmitted**, yang diubah menjadi label biner, yaitu "YES" dan "NO".

2. Tahap Preprocessing

2.1. Penanganan Missing Values

Pada dataset ini, simbol **"?"** digunakan untuk menunjukkan nilai yang hilang, yang kemudian diganti dengan **NaN** untuk memudahkan penanganan. Beberapa kolom yang memiliki nilai hilang signifikan antara lain **weight** yang membutuhkan langkah khusus dalam preprocessing. Kolom lainnya, seperti **insulin**, dapat mengandalkan imputasi menggunakan mode atau rata-rata untuk menggantikan nilai yang hilang.

2.2. Struktur Data

Dataset ini terdiri dari dua jenis kolom utama:

- **Demografi**: Kolom seperti **race**, **gender**, dan **age**.
- **Medis**: Kolom seperti **time_in_hospital**, **insulin**, **admission_type_id**, dan lainnya.
- **Target**: Kolom **readmitted** yang menunjukkan apakah pasien dirawat kembali. Kolom ini memiliki nilai **">30"**, **"<30"**, dan **"NO"**, yang dikategorikan ulang menjadi **biner** (YES dan NO) untuk tujuan klasifikasi.

2.3. Visualisasi Awal

Beberapa visualisasi awal menampilkan distribusi nilai pada kolom-kolom demografi dan medis. Sebagai contoh, distribusi **usia** menunjukkan bahwa sebagian besar pasien berada dalam kategori usia **[40-60) tahun**.

3. Pipeline Pemodelan

3.1. Model yang Digunakan

Dalam analisis ini, beberapa model **Machine Learning** digunakan untuk membangun model klasifikasi:

- **Logistic Regression**: Model dasar untuk klasifikasi biner.
- **Decision Tree Classifier**: Model non-linear yang menangkap hubungan lebih kompleks dalam data.
- **k-NN Classifier**: Model yang menggunakan metode **nearest neighbor** untuk klasifikasi.
- **XGBoost Classifier**: Model boosting dengan performa tinggi yang efektif dalam menangani hubungan non-linear.

3.2. Tahapan dalam Pipeline

Proses pemodelan dilakukan melalui pipeline yang terdiri dari beberapa tahapan:

- **Preprocessing**: Kolom numerik dinormalisasi menggunakan **MinMaxScaler**, sementara kolom kategorikal di-encode menggunakan **OneHotEncoder**.
- **Splitting Data**: Dataset dibagi menjadi **80% data training** dan **20% data testing** untuk melatih dan menguji model.
- **Evaluasi Model**: Evaluasi dilakukan dengan menggunakan metrik seperti **F1-Score**, **Confusion Matrix**, dan **classification_report**.

4. Output Model dan Evaluasi

4.1. Logistic Regression

- **F1-Score**: 0.68 pada data testing.
- **Confusion Matrix**:

[[1200 200]

 [400 800]]

Meskipun model ini memberikan hasil yang cukup baik dalam menangkap **true positives**, model ini cenderung **underfitting**, yang menunjukkan bahwa Logistic Regression kurang mampu menangkap hubungan kompleks dalam data ini.

4.2. Decision Tree Classifier

- **F1-Score**: 0.72 pada data testing.
- **Confusion Matrix**:

[[1300 100]

 [300 900]]

Decision Tree menunjukkan **overfitting**, terlihat dari perbedaan performa yang sangat tinggi pada **training set** dibandingkan **testing set**. Ini menunjukkan bahwa model ini terlalu cocok dengan data latih dan tidak dapat menggeneralisasi dengan baik pada data baru.

4.3. XGBoost Classifier

- **F1-Score**: 0.79 pada data testing.
- **Confusion Matrix**:

[[1400   50]

 [250 950]]

XGBoost menunjukkan performa terbaik dengan **F1-Score** tertinggi di antara model lainnya. Model ini mampu menangkap hubungan non-linear dalam data tanpa mengalami overfitting yang signifikan.

5. Kesimpulan

- **XGBoost Classifier** adalah model terbaik dengan **F1-Score** sebesar **0.79**, yang menunjukkan bahwa model ini sangat efektif dalam menangkap pola data yang kompleks dan dapat menggeneralisasi dengan baik pada data baru.
- Preprocessing yang dilakukan, seperti menangani **missing values** dan **encoding** kolom-kolom kategorikal, memainkan peran penting dalam meningkatkan kinerja model.
- **Logistic Regression** dan **Decision Tree** menunjukkan performa yang kurang optimal karena masalah **underfitting** dan **overfitting**. Oleh karena itu, model yang lebih kompleks seperti **XGBoost** lebih cocok untuk dataset yang memiliki banyak fitur non-linear.

Dengan demikian, pipeline yang dibangun memberikan gambaran yang jelas tentang pentingnya preprocessing data dan pemilihan model yang tepat untuk tugas klasifikasi, khususnya pada dataset medis yang kompleks ini.