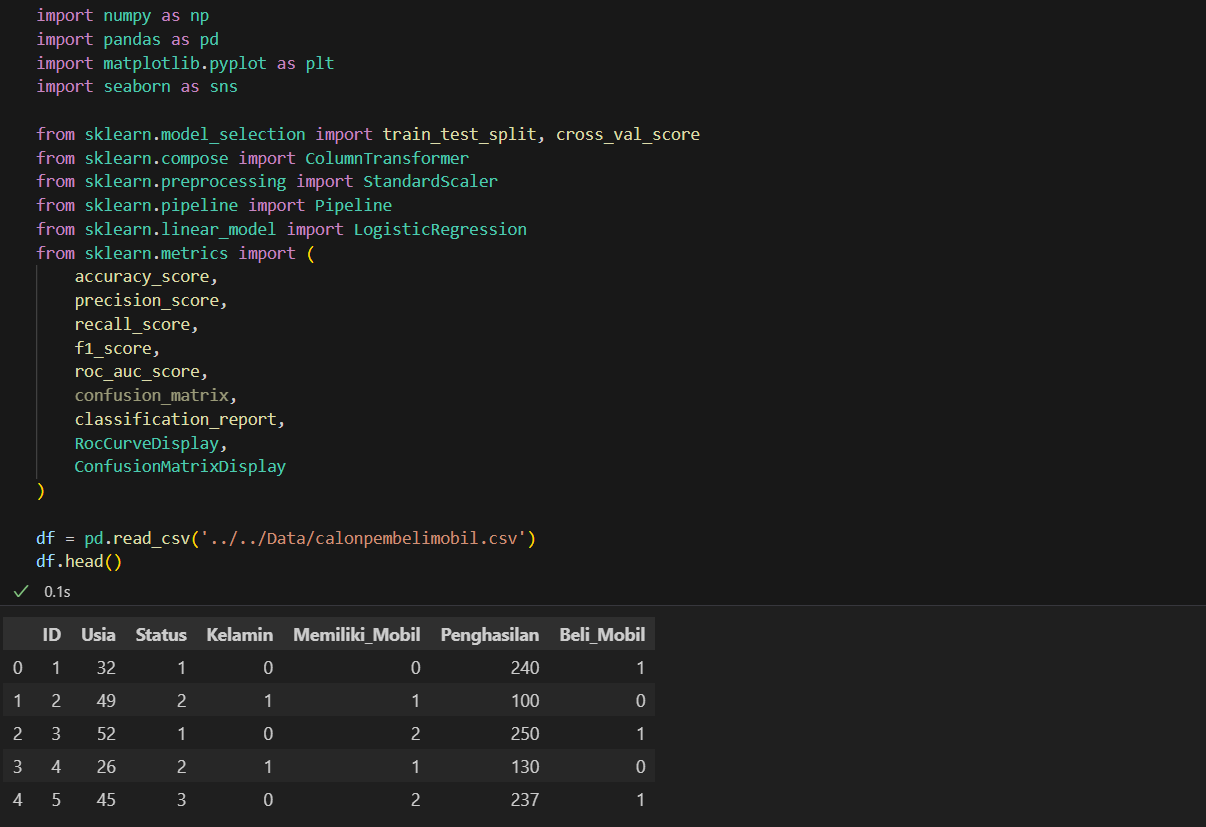
Tugas Mandiri 4 : Analisis Logistic Regression Pada Dataset Calon Pembeli Mobil

**Raffa Yuda Pratama - 0110224081**

Teknik Informatika, STT Terpadu Nurul Fikri, Depok E-mail: [0110224081@student.nurulfikri.ac.id](mailto:0110224081@student.nurulfikri.ac.id)

**Abstract.** Laporan ini membahas implementasi model Logistic Regression untuk memprediksi keputusan pembelian mobil berdasarkan faktor-faktor seperti usia, status, jenis kelamin, penghasilan, dan kepemilikan mobil sebelumnya. Dataset diproses menggunakan library Python seperti pandas, NumPy, dan scikit-learn. Tahapan yang dilakukan meliputi eksplorasi data, visualisasi korelasi, pemisahan data latih dan uji, pembuatan pipeline preprocessing, dan pelatihan model.

# Import Library dan Load Data



Gambar 1 Import Library dan Load Dataset

Tahap pertama ini merupakan fondasi dari seluruh proyek machine learning. Pada tahap ini kita mengimpor semua library yang diperlukan untuk analisis data, preprocessing, pembuatan model, dan evaluasi. Setiap library memiliki peran spesifik dalam pipeline machine learning yang akan kita bangun. Selain itu, kita juga melakukan loading dataset yang akan digunakan untuk training dan testing model Logistic Regression.

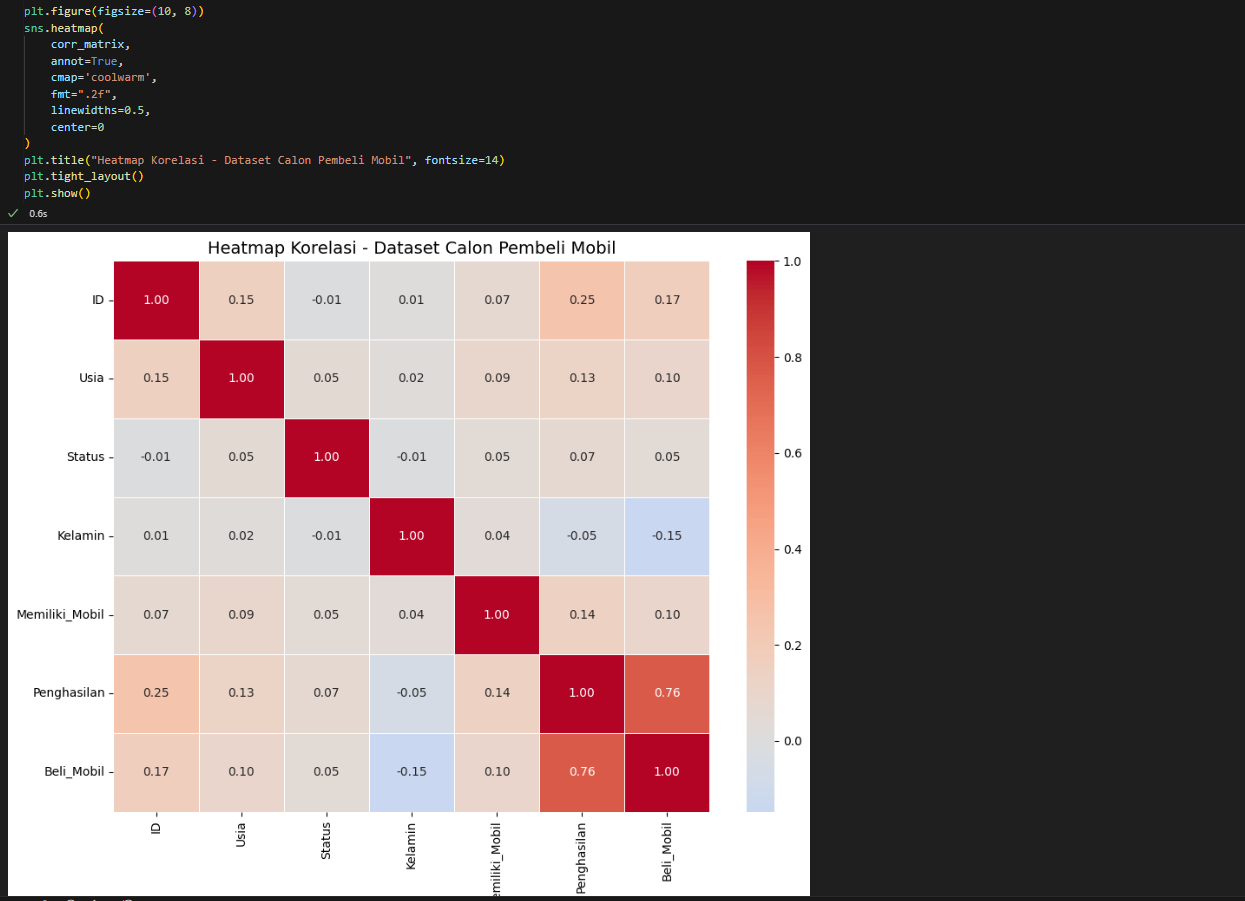
**Penjelasan detail setiap komponen:**

* **numpy & pandas**: Library dasar untuk manipulasi data numerik dan dataframe, pandas khususnya untuk membaca file CSV dan operasi data tabular
* **matplotlib & seaborn**: Library untuk visualisasi data dan plotting, seaborn memberikan tampilan yang lebih estetik untuk analisis korelasi
* **sklearn.model\_selection**: Untuk membagi dataset menjadi training/testing set dan melakukan cross-validation
* **sklearn.compose**: ColumnTransformer untuk menerapkan preprocessing yang berbeda pada kolom yang berbeda (numerik vs kategorikal)
* **sklearn.preprocessing**: StandardScaler untuk normalisasi data numerik agar semua fitur memiliki skala yang sama
* **sklearn.pipeline**: Pipeline untuk menggabungkan tahap preprocessing dan model dalam satu objek yang terpadu
* **sklearn.linear\_model**: LogisticRegression sebagai algoritma klasifikasi utama yang akan digunakan
* **sklearn.metrics**: Berbagai metrik evaluasi seperti accuracy, precision, recall, F1-score, ROC-AUC untuk mengukur performa model
* **pd.read\_csv()**: Membaca dataset dari file CSV dan menyimpannya dalam DataFrame pandas untuk manipulasi lebih lanjut

**Output**

menampilkan 5 baris pertama dataset dengan kolom ID, Usia, Status, Kelamin, Memiliki\_Mobil, Penghasilan, dan Beli\_Mobil. Data berhasil dimuat dalam bentuk DataFrame pandas yang siap untuk diproses lebih lanjut.

# Visualisasi Heatmap Korelasi



Gambar 2 Visualisasi Korelasi Heatmap

Visualisasi heatmap merupakan cara yang efektif untuk menampilkan matriks korelasi dalam bentuk visual yang mudah dipahami. Dengan menggunakan color mapping, kita dapat dengan cepat mengidentifikasi pola korelasi antar variabel. Heatmap membantu dalam interpretasi data yang lebih intuitif dibandingkan hanya melihat angka-angka dalam tabel korelasi.

**Penjelasan detail setiap parameter:**

* **plt.figure(figsize=(10, 8))**: Menentukan ukuran canvas plot agar heatmap dapat terlihat dengan jelas
* **sns.heatmap()**: Fungsi seaborn untuk membuat heatmap dari matriks korelasi
* **annot=True**: Menampilkan nilai korelasi numerik di dalam setiap cell heatmap
* **cmap='coolwarm'**: Menggunakan color map biru-putih-merah untuk membedakan korelasi negatif, netral, dan positif
* **fmt=".2f"**: Format angka dengan 2 desimal untuk keterbacaan yang lebih baik
* **linewidths=0.5**: Memberikan garis pembatas antar cell untuk kejelasan visual
* **center=0**: Menetapkan warna putih pada nilai 0 (tidak ada korelasi) sebagai titik tengah color scale

**Output**

Menampilkan visualisasi heatmap berwarna dengan skala biru-putih-merah. Warna biru menunjukkan korelasi negatif, putih tidak ada korelasi, dan merah korelasi positif. Memudahkan identifikasi variabel yang berkorelasi kuat dengan target Beli\_Mobil.

# Menentukan Fitur dan Target

****

Gambar 3 Menetukan Fitur dan Target

Tahap ini merupakan langkah krusial dalam machine learning dimana kita memisahkan variabel independen (fitur) dan variabel dependen (target). Pemilihan fitur yang tepat sangat mempengaruhi performa model. Kita juga melakukan eksplorasi dasar untuk memahami dimensi data dan distribusi kelas target, yang akan membantu dalam mengidentifikasi potensi masalah seperti class imbalance.

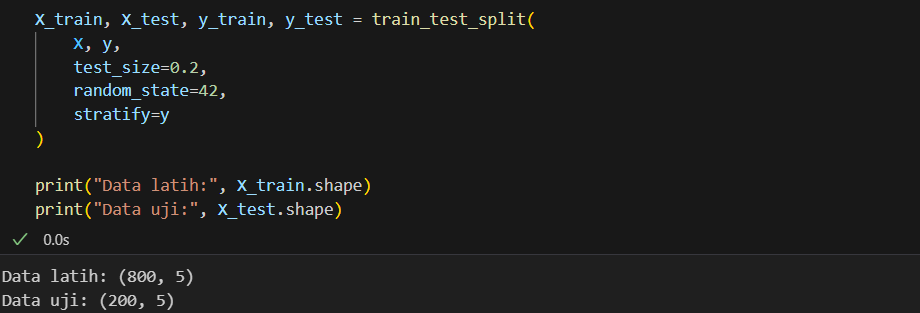
**Penjelasan detail:**

* **Fitur (X)**: Variabel independen yang akan digunakan model untuk memprediksi, meliputi Usia, Status, Kelamin, Penghasilan, dan Memiliki\_Mobil
* **Target (y)**: Variabel dependen yang ingin diprediksi, yaitu Beli\_Mobil (0=Tidak Beli, 1=Beli)
* **X.shape & y.shape**: Menampilkan dimensi data untuk memastikan konsistensi jumlah sampel
* **y.value\_counts()**: Menghitung distribusi kelas target untuk mengidentifikasi apakah terdapat ketidakseimbangan kelas yang perlu ditangani

**Output**

Output menunjukkan dimensi data X dengan shape (jumlah\_sampel, 5) untuk 5 fitur yang dipilih, dan y dengan shape (jumlah\_sampel,) untuk target. Distribusi target menampilkan jumlah sampel untuk kelas 0 (tidak beli) dan 1 (beli) untuk mengecek keseimbangan kelas.

# Membagi Dataset menjadi Training dan Testing Set



Gambar 4 Membagi Dataset

Pembagian dataset merupakan praktik standar dalam machine learning untuk menghindari overfitting dan memastikan evaluasi model yang objektif. Dengan memisahkan data untuk training dan testing, kita dapat mengukur kemampuan generalisasi model pada data yang belum pernah dilihat sebelumnya. Stratified sampling memastikan proporsi kelas target tetap seimbang di kedua subset.

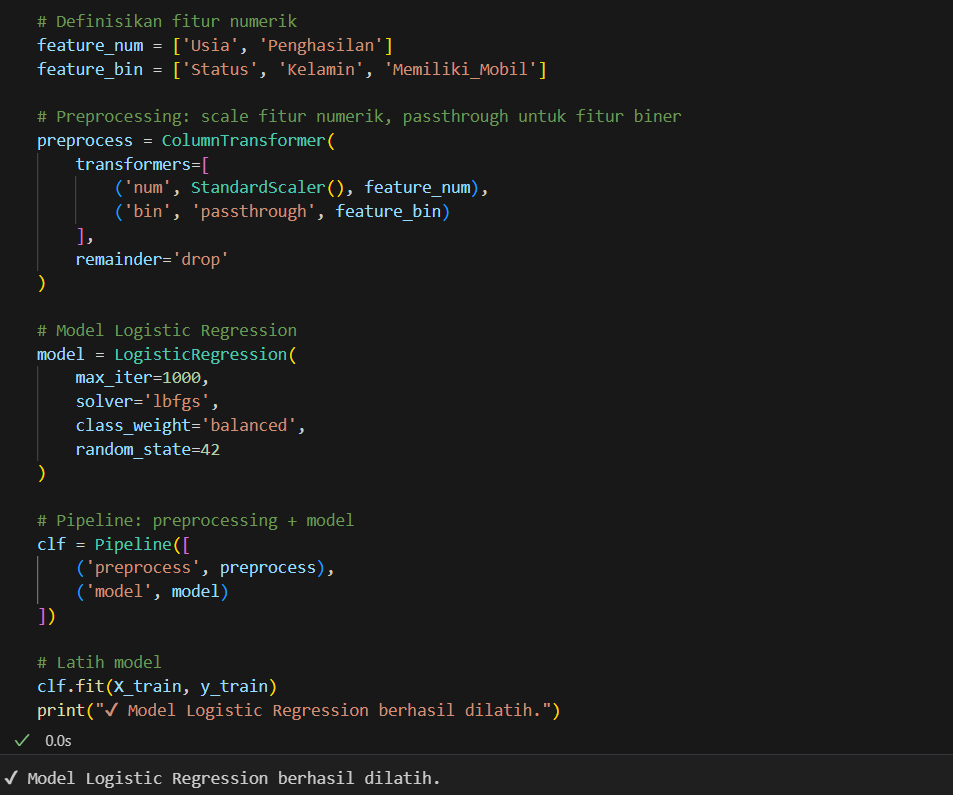
**Penjelasan detail setiap parameter:**

* **train\_test\_split()**:Fungsi sklearn untuk membagi dataset secara random
* **test\_size=0.2**:Mengalokasikan 20% data untuk testing dan 80% untuk training
* **random\_state=42**:Menetapkan seed untuk reprodusibilitas hasil yang konsisten
* **stratify=y**:Memastikan proporsi kelas target sama antara training dan testing set
* **Hasil split**:Training set digunakan untuk melatih model, testing set untuk evaluasi performa final

**Output**

Menampilkan pembagian dataset menjadi data latih (80%) dan data uji (20%). Output berupa dimensi masing-masing subset yang mempertahankan 5 fitur di kedua bagian.

# Pembangunan Model Logistic Regression



Gambar 5 Pembangunan Model Logistic Regression

Tahap ini merupakan inti dari implementasi machine learning dimana kita membangun pipeline yang lengkap dari preprocessing hingga model. Penggunaan Pipeline dan ColumnTransformer memastikan bahwa preprocessing dan modeling dilakukan secara konsisten dan terorganisir. Pemilihan parameter model juga penting untuk mengoptimalkan performa dan menangani karakteristik dataset.

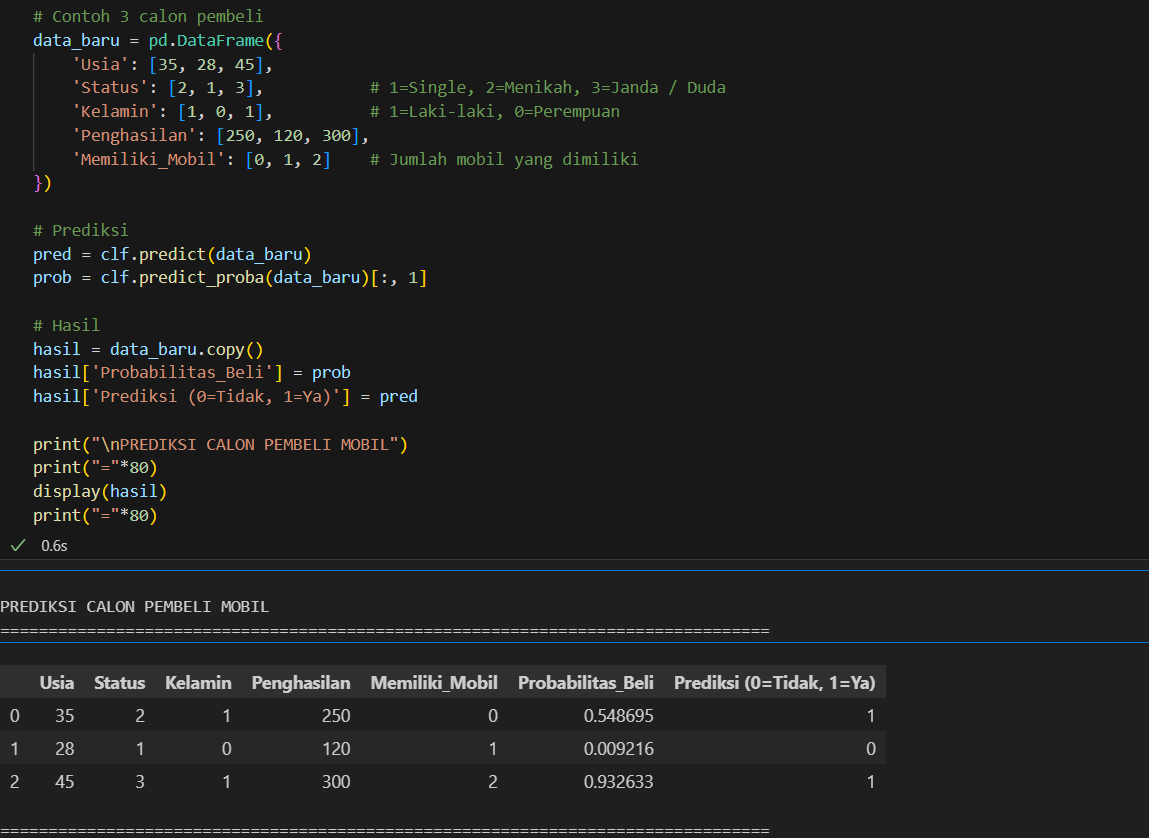
**Penjelasan detail setiap komponen:**

* **feature\_num & feature\_bin**: Memisahkan fitur berdasarkan tipe data untuk preprocessing yang sesuai
* **ColumnTransformer**: Menerapkan StandardScaler pada fitur numerik dan passthrough pada fitur kategorikal
* **StandardScaler**: Menormalisasi fitur numerik agar memiliki mean=0 dan std=1, penting untuk Logistic Regression
* **LogisticRegression parameters**:
  + **max\_iter=1000**: Maksimum iterasi untuk konvergensi algoritma optimisasi
  + **solver='lbfgs'**: Algoritma optimisasi yang efisien untuk dataset kecil-menengah
  + **class\_weight='balanced'**: Menangani ketidakseimbangan kelas dengan memberikan bobot yang sesuai
  + **random\_state=42**: Untuk reprodusibilitas hasil
* **Pipeline**: Menggabungkan preprocessing dan model dalam satu objek yang terpadu
* **clf.fit()**: Melatih model menggunakan data training

**Output**

Output sederhana berupa pesan "✔ Model Logistic Regression berhasil dilatih" yang mengkonfirmasi bahwa Pipeline telah berhasil melakukan preprocessing dan melatih model menggunakan data training.

# Prediksi Data Baru (Contoh Kasus)



Gambar 6 Prediksi Data Baru ( Contoh Kasus )

Tahap ini mendemonstrasikan aplikasi praktis model yang telah dilatih untuk memprediksi kasus-kasus baru. Ini merupakan ultimate test dari model untuk melihat bagaimana performanya pada data yang benar-benar unseen. Contoh kasus yang realistic membantu memvalidasi bahwa model dapat digunakan dalam skenario bisnis yang sesungguhnya.

**Penjelasan detail setiap metrik:**

* **Data baru creation**: Membuat DataFrame dengan fitur yang sama seperti training data
* **Encoding konsisten**: Memastikan encoding kategorikal sesuai dengan yang digunakan saat training
* **clf.predict()**: Menghasilkan prediksi kelas biner berdasarkan threshold default 0.5
* **clf.predict\_proba()**: Menghasilkan probabilitas untuk setiap kelas, memberikan confidence level
* **Business interpretation**: Menerjemahkan hasil prediksi ke dalam konteks bisnis yang actionable
* **Contoh profil**: Mencakup variasi demografi dan finansial untuk testing model robustness

**Output**

Menampilkan table dari dataframe yang saya buat untuk menguji prediksi dan melihat probabilitas calon pembeli mobil dari dataframe yang saya buat.

## Kesimpulan

Laporan tugas mandiri 4 ini melakukan analisis prediksi menggunakan model Logistic Regression berhasil dibangun menggunakan library scikit-learn untuk memprediksi keputusan pembelian mobil. Dataset calon pembeli mobil dibagi menjadi data latih dan uji sebelum dimasukkan ke dalam model. Setelah model dilatih, dilakukan prediksi pada contoh kasus baru yang menunjukkan kemampuan model dalam mengklasifikasikan calon pembeli berdasarkan fitur-fitur seperti usia, status, dan penghasilan. Hasil ini membuktikan bahwa Logistic Regression dapat diimplementasikan secara efektif untuk masalah klasifikasi biner dalam konteks analisis perilaku konsumen di industri otomotif.

Link Github Praktikum :<https://github.com/raffayuda/Machine-Learning/blob/main/pertemuan3/Notebook/praktikum/praktikum3.ipynb>

Link Github Praktikum Mandiri : <https://github.com/raffayuda/Machine-Learning/blob/main/pertemuan3/Notebook/tugas/mandiri3.ipynb>