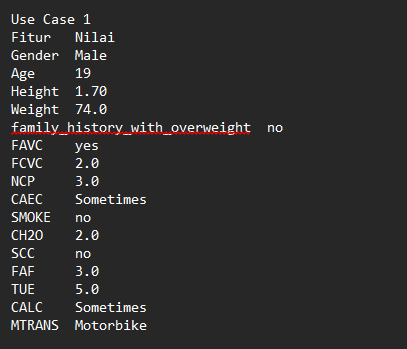
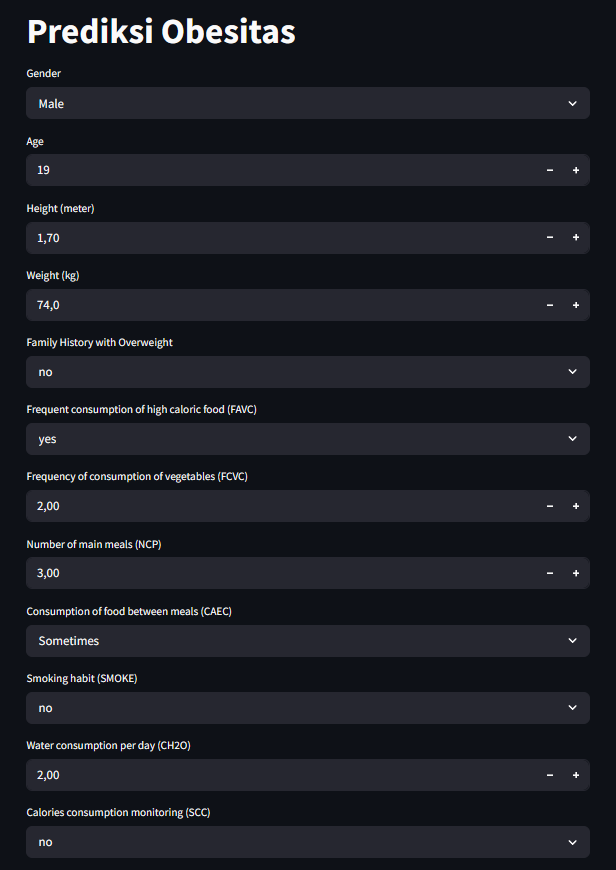
Fidelis Yoga Pradipta

2702296224

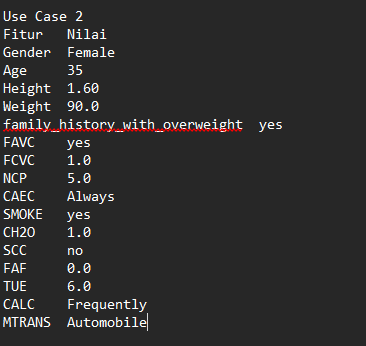
UAS Model Deployment

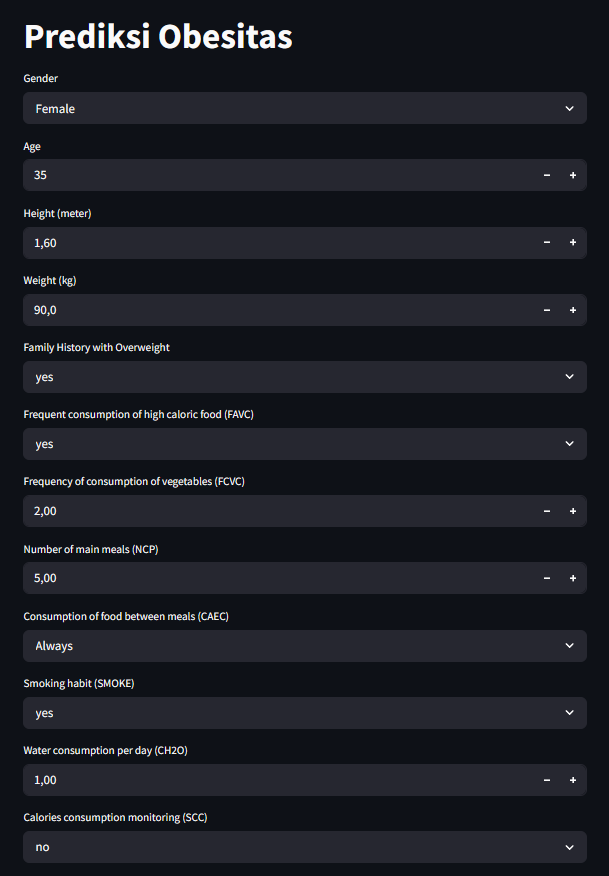
Test Case 1





Test Case 2





**Essay**

1. Testing Model Prediksi Obesitas di Sistem Produksi

Agar model prediksi obesitas bisa berjalan dengan baik saat digunakan di dunia nyata, penting untuk melakukan berbagai pengujian sebelum model benar-benar diluncurkan. Berikut adalah rencana pengujian yang menurut saya perlu dilakukan secara bertahap:

1. Validasi Input

Langkah pertama adalah memastikan semua data input dari pengguna valid dan sesuai format. Misalnya:

* Umur (Age) harus angka positif, bukan string seperti "dua puluh lima".
* Berat badan (Weight) harus masuk akal, antara sekitar 30 sampai 250 kg.
* Kolom seperti SMOKE hanya boleh diisi dengan “yes” atau “no”.
* Untuk MTRANS (transportasi), hanya opsi seperti “Walking”, “Automobile”, atau “Public\_Transportation” yang boleh diterima.

Kalau ada input yang tidak valid, sistem harus bisa langsung memberikan peringatan.

1. Unit Testing

Pengujian ini dilakukan pada bagian-bagian kecil dari sistem, misalnya fungsi prediksi di backend FastAPI. Kita bisa tes apakah fungsi tersebut bisa menangani input yang diberikan, dan apakah hasil prediksinya sesuai.

1. Integration Testing

Setelah unit test, penting juga memastikan frontend (Streamlit) dan backend bisa bekerja sama dengan lancar. Jadi, ketika user isi form, datanya dikirim ke backend, dan hasil prediksi bisa muncul kembali ke user dengan benar.

1. Performance Testing

Model juga perlu dites seberapa cepat bisa memberikan hasil. Idealnya, hasil prediksi bisa keluar kurang dari satu detik, apalagi kalau sistem ini nantinya dipakai oleh banyak orang sekaligus.

1. Adversarial Input Testing

Ini untuk mengetes bagaimana sistem merespons input yang aneh atau dimanipulasi. Misalnya:

* Kalau Age diisi 999 atau -10, sistem seharusnya menolak.
* Kalau SMOKE diisi “maybe” atau MTRANS diisi “Spaceship”, itu juga harus ditolak karena bukan data valid.

1. Monitoring dan Logging

Terakhir, sistem harus mencatat (log) semua request dan hasilnya. Ini penting untuk analisis jika suatu saat terjadi error atau performa model menurun.

1. Model drift dan Data drift

Dalam penerapan model machine learning di dunia nyata kita sering menemui situasi di mana akurasi model menurun setelah beberapa waktu digunakan. Hal ini bisa disebabkan oleh dua hal penting yaitu data drift dan model drift.

Data drift adalah kondisi di mana data input yang masuk ke sistem mengalami perubahan distribusi dibandingkan dengan data saat model pertama kali dilatih. Jadi meskipun modelnya tidak berubah, data yang masuk sudah berbeda dan ini bisa membuat prediksi model menjadi kurang akurat. Contohnya saat model prediksi obesitas pertama kali dilatih, sebagian besar pengguna menggunakan mode transportasi seperti "Motorbike" dan "Walking". Namun, setelah beberapa bulan, makin banyak pengguna yang memakai "Public Transportation" atau "Bike". Perubahan ini menyebabkan model bingung, karena pola baru ini tidak dikenali dari data training awal. Jadi, data drift terjadi karena perubahan pada data input. Modelnya tetap sama tapi performanya turun karena input yang sekarang sudah berbeda.

Model drift terjadi ketika hubungan antara fitur (input) dan target (label) sudah berubah seiring waktu. Jadi, walaupun input dan labelnya secara bentuk masih sama, hubungan logis di antara keduanya tidak lagi sesuai dengan yang dipelajari oleh model saat awal. Dulu orang yang sering makan junk food cenderung pasti obesitas. Tapi sekarang banyak orang mulai rutin olahraga dan menjaga pola hidup sehat, jadi walaupun pola makan sama belum tentu hasil akhirnya tetap obesitas. Karena modelnya dilatih dengan data lama, ia tidak bisa menangkap pola hidup baru ini. Jadi model drift adalah saat model sudah tidak lagi relevan karena pola hubungan dalam data sudah berubah dan solusinya sering kali butuh pelatihan ulang atau bahkan membangun model baru.

Link Gdrive:

<https://drive.google.com/drive/folders/1uR8s6Oyklc3qOZg4KZNrNH3Q0kWK1DlQ?usp=sharing>