

LINK:

<https://drive.google.com/file/d/1orwwaveL1TDLv9x-otPxSPSOONIWL7UD/view?usp=sharing>

LAMPIRAN SS TEST CASE:

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{ "Gender": "Male", "Age": 35, "Height": 1.65, "Weight": 120.0, "family_history_with_overweight": "yes", "FVC": "yes", "FEV1": 1.0, "NCP": 5.0, "CAEC": "Frequently", "SMOKE": "no", "CHZD": 1.0, "SCC": "no", "FAT": 0.0, "TUE": 3.0, "CALC": "Frequently", "MTRANS": "Automobile" }'
```

Request URL

```
http://127.0.0.1:8000/predict
```

Server response

Code	Details
200	Response body

```
{ "prediction": "Obesity_Type_II", "probabilities": { "Insufficient_Weight": 0, "Normal_Weight": 0, "Obesity_Type_I": 0, "Obesity_Type_III": 0, "Obesity_Type_IV": 0, "Overweight_Level_I": 0, "Overweight_Level_II": 0 } }
```

Response headers

```
content-length: 213
content-type: application/json
date: Thu, 19 Jun 2025 08:41:54 GMT
server: uvicorn
```

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{ "Gender": "Male", "Age": 19, "Height": 1.85, "Weight": 60.0, "family_history_with_overweight": "no", "FVC": "no", "FEV1": 1.0, "NCP": 5.0, "CAEC": "Frequently", "SMOKE": "no", "CHZD": 1.0, "SCC": "no", "FAT": 0.0, "TUE": 3.0, "CALC": "Frequently", "MTRANS": "Automobile" }'
```

Request URL

```
http://127.0.0.1:8000/predict
```

Server response

Code	Details
200	Response body

```
{ "prediction": "Normal_Weight", "probabilities": { "Insufficient_Weight": 0.0011, "Normal_Weight": 0.9989, "Obesity_Type_I": 0, "Obesity_Type_III": 0, "Obesity_Type_IV": 0, "Overweight_Level_I": 0, "Overweight_Level_II": 0 } }
```

Response headers

```
content-length: 217
content-type: application/json
date: Thu, 19 Jun 2025 08:43:48 GMT
server: uvicorn
```

Responses

# Prediksi Tingkat Obesitas

Masukkan data untuk memprediksi tingkat obesitas berdasarkan gaya hidup.

Gender

Male

Age

14

55

Height (m)

1.45

1.45

1.95

Weight (kg)

39.00

173.00

173.00

Family history with overweight?

yes

Frequent consumption of high caloric food?

yes

Vegetable consumption (1-3)

1.00

2.00

3.00

Main meals per day (1-4)

1.00

3.00

4.00

Eating between meals?

No

Do you smoke?

yes

Water intake (1-3 liters)

Vegetable consumption (1-3)

Main meals per day (1-4)

Eating between meals?

No

Do you smoke?

yes

Water intake (1-3 liters)

Do you monitor calorie intake?

yes

Physical activity (0-3)

Time using tech devices (0-2)

Alcohol consumption?

No

Primary transport method

Automobile

Predict

Prediction Result: Obesity\_Type\_III

## Class Probabilities

	Probability
Insufficient_Weight	0
Normal_Weight	0
Obesity_Type_I	0
Obesity_Type_II	0.1502
Obesity_Type_III	0.8498
Overweight_Level_I	0
Overweight_Level_II	0



# Prediksi Tingkat Obesitas

Masukkan data untuk memprediksi tingkat obesitas berdasarkan gaya hidup.

Gender

Female



Age

33

14

55

Height (m)

1.45

1.45

1.95

Weight (kg)

103.60

39.00

173.00

Family history with overweight?

no



Frequent consumption of high caloric food?

no



Vegetable consumption (1-3)

1.50

1.00

3.00

Main meals per day (1-4)

2.17

1.00

4.00

Eating between meals?

No



Do you smoke?

yes



1.00 3.00

Do you monitor calorie intake?

yes

Physical activity (0-3)

0.00 3.00

2.33

Time using tech devices (0-2)

0.00 2.00

2.00

Alcohol consumption?

No

Primary transport method

Automobile

 Predict

Prediction Result: Obesity\_Type\_III

### Class Probabilities

	Probability
Insufficient_Weight	0
Normal_Weight	0
Obesity_Type_I	0
Obesity_Type_II	0
Obesity_Type_III	1
Overweight_Level_I	0
Overweight_Level_II	0

## ESSAY:

1. Testing yang saya akan lakukan :

- **Functional Testing**

Pengujian ini memastikan bahwa model dan sistem secara keseluruhan berfungsi sebagaimana mestinya. Karena kalau tidak maka prediction tidak akan muncul dan hanya akan ada error yang muncul.

- **Tujuan:** Memastikan input menghasilkan output prediksi obesitas.

Contoh input json untuk Fast api

```
{  
    "Gender": "Male",  
    "Age": 35,  
    "Height": 1.65,  
    "Weight": 120.0,  
    "family_history_with_overweight": "yes",  
    "FAVC": "yes",  
    "FCVC": 1.0,  
    "NCP": 5.0,  
    "CAEC": "Frequently",  
    "SMOKE": "no",  
    "CH2O": 1.0,  
    "SCC": "no",  
    "FAF": 0.0,  
    "TUE": 3.0,  
    "CALC": "Frequently",  
    "MTRANS": "Automobile"  
}
```

OUTPUT:

```
{  
    "prediction": "Obesity_Type_II"  
}
```

- **Validation Testing**

Pengujian ini bertujuan memverifikasi bahwa sistem dapat menangani berbagai variasi nilai input yang sah, sesuai distribusi data saat pelatihan.

Yang saya lakukan adalah membuat min max di streamlit sesuai dengan output dari function yang saya buat di IPYNB

Minimum and Maximum values in Training Dataset:

Age: Min = 14, Max = 55

Height: Min = 1.45, Max = 1.95

Weight: Min = 39.0, Max = 173.0

FCVC: Min = 1.0, Max = 3.0

NCP: Min = 1.0, Max = 4.0

CH2O: Min = 1.0, Max = 3.0

FAF: Min = 0.0, Max = 3.0

TUE: Min = 0.0, Max = 2.0

Dan untuk categorical dtype saya samakan dengan df[col].unique()

- **Contoh pengujian:**

- Age: harus antara 14 dan 55 tahun.
- Weight: harus antara 39.0 dan 173.0 kg.
- SMOKE: hanya menerima "yes" atau "no".
- MTRANS: harus salah satu dari ["Automobile", "Bike", "Motorbike", "Public\_Transportation", "Walking"].

- **Manfaat:** Mencegah input tidak realistik yang dapat merusak prediksi.

- **Invalid Input Testing (Robustness Test)**

Pengujian ini memastikan sistem **tahan terhadap input tidak valid**, salah ketik, atau manipulasi input oleh user.

**Contoh Kasus Manipulasi:**

Fitur	Contoh Manipulasi	
Age	-1, 200, "dua puluh"	
Weight	null, "seratus kilogram"	
SMOKE	"maybe", "0"	
MTRANS	"Teleportasi", angka 1	

- Sistem harusnya tidak crash, tapi menolak input
- Contoh di JSON fast api saya ubah input -> age “40” jadi “empatpuluhan”
- Maka akan error dengan code 422 dan detail “Error: Unprocessable Entity”

```
Curl
curl -X 'POST' \
  'http://127.0.0.1:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "Gender": "Female",
    "Age": "empatpuluhan",
    "Height": 160,
    "Weight": 45.0,
    "family_history_with_overweight": "yes",
    "HCV": "no",
    "ECV": "no",
    "NCP": 5.0,
    "NCK": "Frequently",
    "SNCK": "No",
    "QHQ": 1.0,
    "PCC": "no",
    "PBC": 0.0,
    "PUE": 3.0,
    "CALC": "No",
    "MTRANS": "Automobile"
  }'

Request URL
http://127.0.0.1:8000/predict

Server response
Code Details

422 Error: Unprocessable Entity
Response body
{
  "detail": [
    {
      "type": "json_invalid",
      "loc": [
        "body"
      ],
      "msg": "JSON decode error",
      "inner": {}
    }
  ]
}

Download

Response headers
content-length: 125
content-type: application/json
date: Thu, 19 Jun 2025 07:30:35 GMT
server: unicorn

Responses
Code Description Links
```

- **Performance Testing**

Pengujian ini menilai kecepatan, skalabilitas, dan efisiensi model.

- **Tujuan:** Memastikan model dapat menangani beban kerja di lingkungan produksi tanpa degradasi performa yang signifikan.
- **Contoh Implementasi:**
  - **Latency:** Ukur waktu yang dibutuhkan model untuk menghasilkan prediksi tunggal atau batch.
  - **Throughput:** Uji berapa banyak prediksi yang dapat diproses model per detik/ per menit di bawah berbagai beban pengguna bersamaan.
  - **Resource Utilization:** Pantau penggunaan CPU, memori, dan GPU saat model berjalan. Identifikasi potensi *bottleneck*.

- **Recovery Testing**

Pengujian ini memverifikasi kemampuan sistem untuk pulih dari kegagalan.

- Tujuan: Memastikan model dan infrastruktur pendukung dapat pulih dengan cepat setelah kegagalan.
- Contoh Implementasi:
  - Simulasikan kegagalan server tempat model berjalan atau hilangnya koneksi database. Perhatikan apakah model dapat secara otomatis memulai ulang atau beralih ke *instance* cadangan.

- **Security Testing**

Pengujian ini berfokus pada kerentanan model terhadap serangan atau manipulasi yang tidak sah.

- **Tujuan:** Memastikan model aman dari ancaman eksternal.
- Contoh Implementasi:
  - **Serangan Adversarial:** Coba berikan input yang sedikit dimodifikasi yang dirancang untuk membingungkan model dan menghasilkan prediksi yang salah (misalnya, memanipulasi Age atau Weight sedikit untuk mengubah prediksi dari 'Normal Weight' menjadi 'Obesity Type III' atau sebaliknya).
  - **Injeksi Data Malicious:** Uji apakah sistem dapat menahan injeksi data yang bertujuan untuk merusak model atau mengganggu operasinya.

## 2. DATA DRIFT DAN MODEL DRIFT

**Data Drift:** perubahan distribusi data input dari waktu ke waktu. Ini berarti karakteristik data baru yang diterima model di produksi berbeda secara signifikan dari data yang digunakan untuk melatih model awalnya.

**Dalam konteks sistem prediksi obesitas kita:**

model dilatih pada data historis pasien dari beberapa tahun yang lalu. Seiring waktu, kebiasaan hidup, pola makan, atau faktor demografi populasi pasti akan berubah seiring perkembangan zaman.

- **Contoh Data Drift:**
  - **Perubahan Rata-rata/Standar Deviasi:** Rata-rata Weight atau Age pasien yang datang ke klinik mungkin meningkat secara bertahap seiring waktu. Jika model dilatih pada populasi dengan rata-rata Weight 70 kg, tetapi sekarang pasien rata-rata 80 kg, maka prediksi model mungkin kurang akurat karena *range* data Weight yang dilihat model sudah bergeser.
  - **Perubahan Distribusi Fitur Kategorikal:** Proporsi MTRANS (Moda transportasi utama) mungkin bergeser dari dominan "Transportasi Umum" menjadi lebih banyak "Mobil" karena peningkatan kepemilikan kendaraan pribadi. Jika model sangat bergantung pada fitur ini, pergeseran distribusi dapat mengurangi akurasi model kita.
  - **Perubahan dalam FAVC (Frekuensi konsumsi makanan berkalori tinggi):** Seiring waktu, tren diet atau ketersediaan makanan cepat saji dapat mengubah frekuensi FAVC dalam populasi.
  - **Perubahan dalam korelasi antar fitur:** Mungkin di data pelatihan, SMOKE dan CALC (konsumsi alkohol) sangat berkorelasi. Namun, seiring waktu, korelasi ini mungkin melemah atau menguat karena perubahan gaya hidup, yang pada akhirnya memengaruhi bagaimana model memanfaatkan fitur-fitur tersebut.
- Intinya ada perubahan dari pola hidup sehingga tren data sekarang berubah dari tren data pada saat training (data historis yang dipakai). Adanya perubahan distribusi input
- Solusi: Analisis distribusi & retraining

**Model Drift:** terjadi ketika hubungan antara data input dan target output berubah seiring waktu. Artinya, bahkan jika distribusi data input tetap sama, definisi dari 'obesitas' itu sendiri, atau bagaimana fitur-fitur memengaruhi obesitas, telah berubah. Model yang dulunya akurat gak lagi karena "konsep" yang dipelajarinya telah usang.

Model bisa outdated karena:

- Tidak update dengan pola baru
- Terlalu overfitting ke data lama
- Terjadi perubahan gaya hidup penduduk secara gradual

Dalam Konteks Sistem Prediksi Obesitas:

- **Contoh Model Drift (Concept Drift):**
  - **Perubahan Definisi Obesitas:** Meskipun *outcome* NObeyesedad (tingkat obesitas) memiliki kategori yang jelas, faktor-faktor yang menyebabkan seseorang masuk ke kategori tersebut mungkin berubah. Misalnya, karena adanya pemahaman medis baru atau perubahan gaya hidup, kombinasi Age, Weight, Height, dan FAVC yang dulunya memprediksi "Overweight Level I" sekarang mungkin lebih cenderung memprediksi "Normal Weight" atau "Obesity Type I".
  - **Perubahan Dampak Fitur:** Dulu, SMOKE mungkin memiliki dampak signifikan terhadap risiko obesitas pada suatu populasi, tetapi sekarang, karena faktor gaya hidup lain yang lebih dominan, dampak SMOKE menjadi lebih kecil. Model yang dilatih pada hubungan lama akan menjadi kurang akurat.
  - **Interaksi Fitur Baru:** Mungkin ada interaksi baru antara TUE (waktu penggunaan teknologi) dan FAF (frekuensi aktivitas fisik) yang tidak ada atau tidak signifikan pada saat model dilatih, tetapi sekarang menjadi penting dalam memprediksi obesitas. Model lama tidak akan menangkap interaksi ini.
- Intinya perubahan pada hubungan antara input dan output
- Solusi: evaluasi ulang arsitektur/model