import numpy as np

import pickle

# Load the trained scaler

scaler\_filename = 'scaler.sav'

scaler = pickle.load(open(scaler\_filename, 'rb'))

# Load the trained classifier

classifier\_filename = 'classifier.sav'

classifier = pickle.load(open(classifier\_filename, 'rb'))

# Data input

input\_data = (19.8, 0, -2.8, 5)

# Process the input data

input\_data\_as\_numpy\_array = np.array(input\_data)

input\_data\_reshape = input\_data\_as\_numpy\_array.reshape(1, -1)

# Standardize the data

std\_data = scaler.transform(input\_data\_reshape)

# Make the prediction

prediction = classifier.predict(std\_data)

# Interpret the prediction result

kemungkinan = prediction[0]

if kemungkinan == 0:

print("Aman untuk keluar.")

elif kemungkinan == 1:

print("Aman untuk keluar, tapi perhatikan dengan seksama.")

elif kemungkinan == 2:

print("Tidak disarankan untuk keluar.")

elif kemungkinan == 3:

print("Sangat tidak aman untuk keluar!")

else:

ini yang lama

import pickle

import streamlit as st

# Membaca model yang telah dilatih

prediction\_model = pickle.load(open('prediksi\_cuaca.sav', 'rb'))

# Judul halaman web

st.title('Prediksi Cuaca')

# Membagi kolom untuk input data

col1, col2 = st.columns(2)

# Input data numerik

with col1:

    precipitation = st.number\_input('Input Nilai Presipitasi', min\_value=0.0, step=0.1)

    temp\_max = st.number\_input('Input Suhu Maksimum', min\_value=-100.0, max\_value=50.0)

with col2:

    temp\_min = st.number\_input('Input Suhu Minimum', min\_value=-100.0, max\_value=50.0)

    wind = st.number\_input('Input Kecepatan Angin', min\_value=0.0, step=0.1)

# Variabel untuk menyimpan hasil prediksi

weather\_prediction = ''

# Tombol untuk melakukan prediksi

if st.button('Lihat Hasil Prediksi'):

    # Prediksi cuaca berdasarkan input data

    weather\_prediction = prediction\_model.predict([[precipitation, temp\_max, temp\_min, wind]])

    # Interpretasi hasil prediksi

    if weather\_prediction[0] == 0:

        weather\_prediction = 'Aman untuk keluar rumah.'

    elif weather\_prediction[1] == 0:

        weather\_prediction = 'Aman untuk keluar rumah.'

    elif weather\_prediction[2] == 0:

        weather\_prediction = 'Tidak aman untuk keluar rumah.'

    elif weather\_prediction[3] == 0:

        weather\_prediction = 'Tidak aman untuk keluar rumah.'

    else:

        weather\_prediction = 'Terjadi kesalahan pada model. Silakan periksa kembali model Anda.'

    # Menampilkan hasil prediksi

    st.success(weather\_prediction)