

```

import streamlit as st
import pandas as pd
import pickle

# Load the trained model
filename = r'knn_model.sav'
loaded_model = pickle.load(open(filename, 'rb'))

# Define the correct column names
columns = ['Delivery_Distance', 'Traffic_Congestion', 'Weather_Condition', 'Delivery_Slot', 'Driver_Experience', 'Num_Stops', 'Vehicle_Age', 'Road_Condition_Score', 'Package_Weight', 'Fuel_Efficiency', 'Warehouse_Processing_Time']

# Define the prediction function
def predict_delivery_delay(features):
    """ Predicts the delivery delay based on input features. """
    prediction = loaded_model.predict(features)
    return prediction

# Create the Streamlit app
st.title("Delivery Delay Prediction")

# Get user input
st.write("Please provide the following information:")

Delivery_Distance = st.number_input("Delivery Distance (in km)", min_value=0.0)
Traffic_Congestion = st.number_input("Traffic Congestion Level (1-5)", min_value=1, max_value=5)
Weather_Condition = st.number_input("Weather Condition (1-5)", min_value=1, max_value=5)
Delivery_Slot = st.number_input("Delivery Slot (1-based index)", min_value=1)
Driver_Experience = st.number_input("Driver Experience (in years)", min_value=0.0)
Num_Stops = st.number_input("Number of Stops", min_value=0)
Vehicle_Age = st.number_input("Vehicle Age (in years)", min_value=0.0)
Road_Condition_Score = st.number_input("Road Condition Score (1-5)", min_value=1, max_value=5)
Package_Weight = st.number_input("Package Weight (in kg)", min_value=0.0)
Fuel_Efficiency = st.number_input("Fuel Efficiency (in km/liter)", min_value=0.0)
Warehouse_Processing_Time = st.number_input("Warehouse Processing Time (in minutes)", min_value=0.0)

# Create a dataframe with the user input
input_data = pd.DataFrame([[Delivery_Distance, Traffic_Congestion, Weather_Condition, Delivery_Slot, Driver_Experience, Num_Stops, Vehicle_Age, Road_Condition_Score, Package_Weight, Fuel_Efficiency, Warehouse_Processing_Time]], columns=columns)

# Make a prediction
# Make a prediction if
st.button("Predict Delivery Delay"):
    prediction = predict_delivery_delay(input_data)
    if prediction[0] == 0:
        st.write("Predicted Delivery Delay: 0 (No significant delay expected)")
    else:
        st.write("Predicted Delivery Delay: 1 (Delay expected)")

```