

EXECUTABLE CODE

Train. model :

```
import pandas as pd
```

```
import joblib
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.linear_model import LinearRegression
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.preprocessing import LabelEncoder
```

```
# Load dataset with correct encoding
```

```
file_path = "dataset/lpg_india_dataset_250.csv"
```

```
df = pd.read_csv(file_path, encoding='utf-8') # Change encoding if needed
```

```
# Print column names to check formatting issues
```

```
print("Original column names:", df.columns.tolist())
```

```
# Rename columns to remove any encoding issues or spaces
```

```
df.columns = df.columns.str.replace(r"^[^x00-\x7F]+", "", regex=True) # Remove special characters
```

```
df.columns = df.columns.str.strip() # Remove extra spaces
```

```
# Print column names after cleaning
```

```
print("Updated column names:", df.columns.tolist())
```

```
# Define features (X) and target variables (y)
```

```
X = df[['Tare Weight (kg)', 'LPG Weight (kg)', 'Gross Weight (kg)', 'Temperature (C)', 'Humidity (%)', 'Air Pressure (hPa)']]
```

```
# Target 1: LPG Percentage Prediction
```

```
y_percentage = (df['LPG Weight (kg)'] / df['Gross Weight (kg)']) * 100 # Compute LPG % dynamically
```

```
# Target 2: Status Prediction (Convert categories into numbers)
```

```
label_encoder = LabelEncoder()
```

```
df['Status'] = label_encoder.fit_transform(df['Status']) # Convert 'Safe', 'Low', 'Critical' to numbers
```

```
y_status = df['Status']
```

```
# Split data into training and testing sets
```

```
X_train, X_test, y_percentage_train, y_percentage_test = train_test_split(X, y_percentage,  
test_size=0.2, random_state=42)
```

```
X_train, X_test, y_status_train, y_status_test = train_test_split(X, y_status, test_size=0.2,  
random_state=42)
```

```
# Train LPG Percentage Model (Regression)
```

```
model_percentage = LinearRegression()
```

```
model_percentage.fit(X_train, y_percentage_train)
```

```
# Train Status Model (Classification)
```

```
model_status = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
model_status.fit(X_train, y_status_train)
```

```
# Save models
```

```
joblib.dump(model_percentage, "models/lpg_percentage_model.pkl")
```

```
joblib.dump(model_status, "models/lpg_status_model.pkl")
```

```
joblib.dump(label_encoder, "models/status_label_encoder.pkl") # Save label encoder for later  
decoding
```

```
print("✅ Models trained and saved successfully!")
```

```
Predict.py
```

```
import pandas as pd
```

```
import joblib
```

```
import numpy as np

# Load model and scaler
model_perc = joblib.load("models/lpg_percentage_model.pkl")
scaler = joblib.load("models/scaler.pkl")

# Function to predict LPG percentage and status
def predict_lpg(tare_weight, lpg_weight, gross_weight, temperature, humidity, air_pressure):
    # Prepare input data
    input_data = np.array([[tare_weight, lpg_weight, gross_weight, temperature, humidity,
air_pressure]])

    # Scale input
    input_scaled = scaler.transform(input_data)

    # Predict LPG Percentage
    lpg_percentage = model_perc.predict(input_scaled)[0]

    # Determine status
    if temperature == 29:
        status = "Safe"
    elif lpg_percentage > 60:
        status = "High"
    elif lpg_percentage > 30:
        status = "Medium"
    else:
        status = "Low"

    return round(lpg_percentage, 2), status

# Example test
```

```

if __name__ == "__main__":
    tare_weight = 15.0
    lpg_weight = 5.0
    gross_weight = 20.0
    temperature = 25.0
    humidity = 50.0
    air_pressure = 1000.0

    percentage, status = predict_lpg(tare_weight, lpg_weight, gross_weight, temperature, humidity,
    air_pressure)

    print(f"Predicted LPG Percentage: {percentage}%")
    print(f"Status: {status}")

```

app.py

```

from flask import Flask, render_template, request
import joblib
import numpy as np

app = Flask(__name__)

# Load trained models
model_percentage = joblib.load("models/lpg_percentage_model.pkl")
model_status = joblib.load("models/lpg_status_model.pkl")

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    try:

```

```

# Get user inputs
tare_weight = float(request.form['tare_weight'])
lpg_weight = float(request.form['lpg_weight'])
gross_weight = float(request.form['gross_weight'])
temperature = float(request.form['temperature'])
humidity = float(request.form['humidity'])
air_pressure = float(request.form['air_pressure'])

# Prepare input for models
input_data = np.array([[tare_weight, lpg_weight, gross_weight, temperature, humidity,
air_pressure]])

# Predict LPG Percentage
lpg_percentage = model_percentage.predict(input_data)[0]

# Predict Status
status_code = int(model_status.predict(input_data)[0]) # Ensure it's an integer

# Convert numeric status to readable text
status_labels = {0: "Safe", 1: "Low", 2: "Critical"}
status_str = status_labels.get(status_code, "Unknown")

return render_template('result.html', percentage=round(lpg_percentage, 2),
status=status_str.lower())

except Exception as e:
    return f"Error: {str(e)}"

if __name__ == '__main__':
    app.run(debug=True)

```