# Import necessary libraries

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, classification\_report

from sklearn.preprocessing import LabelEncoder

# Load the dataset

data = pd.read\_csv('lung\_cancer\_data.csv')

# Data preprocessing

# Handling missing values

data.fillna(method='ffill', inplace=True)

# Encoding categorical variables

label\_encoders = {}

for column in ['gender', 'smoking\_status']:

label\_encoders[column] = LabelEncoder()

data[column] = label\_encoders[column].fit\_transform(data[column])

# Split data into features (X) and target variable (y)

X = data.drop('lung\_cancer', axis=1) # Assuming 'lung\_cancer' is the target variable

y = data['lung\_cancer']

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Model training with adjusted hyperparameters

model = RandomForestClassifier(n\_estimators=200, max\_depth=5, random\_state=42) # Adjusted hyperparameters

model.fit(X\_train, y\_train)

# Model evaluation

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

print("Classification Report:")

print(classification\_report(y\_test, y\_pred))

# Deployment: Saving the trained model

import joblib

joblib.dump(model, 'lung\_cancer\_prediction\_model.pkl')

# Example of making predictions with new data

new\_data = pd.DataFrame([[65, 'male', 'smoker', ...]]) # Example new data

# Preprocess new data similarly to training data (handle missing values, encode categorical variables, etc.)

new\_data[1] = label\_encoders['gender'].transform([new\_data[1]]) # Encoding gender

new\_data[2] = label\_encoders['smoking\_status'].transform([new\_data[2]]) # Encoding smoking status

prediction = model.predict(new\_data)

print("Prediction:", prediction)