# Import libraries

import pandas as pd

from sklearn.preprocessing import LabelEncoder

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

from sklearn.neighbors import KNeighborsRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

# Sample dataset without nulls or blanks

data = {

'Experience': [1, 3, 5, 7, 9, 2, 4, 6, 8, 10],

'Education': ['Bachelor', 'Bachelor', 'Master', 'Master', 'PhD',

'Bachelor', 'Master', 'Master', 'PhD', 'PhD'],

'JobTitle': ['Developer', 'Tester', 'Analyst', 'Manager', 'Director',

'Developer', 'Analyst', 'Manager', 'Director', 'Director'],

'Salary': [300000, 400000, 600000, 900000, 1500000,

320000, 610000, 950000, 1450000, 1550000]

}

df = pd.DataFrame(data)

# check for null or blanks

assert not df.isnull().values.any(), "Dataset contains null values!"

assert not (df.astype(str).apply(lambda x: x.str.strip()) == '').any().any(), "Dataset contains blank strings!"

# Encode categorical features

le\_edu = LabelEncoder()

le\_job = LabelEncoder()

df['Education'] = le\_edu.fit\_transform(df['Education']) # Bachelor=0, Master=1, PhD=2

df['JobTitle'] = le\_job.fit\_transform(df['JobTitle']) # Developer=0, etc.

# Features and Target

X = df[['Experience', 'Education', 'JobTitle']]

y = df['Salary']

# Split the dataset

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

# Initialize and train KNN model

model = KNeighborsRegressor(n\_neighbors=3)

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

# Predict and evaluate

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

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)