Centre for infrastructure, Sustainable

Transportation and Urban Planning

Indian Institute of Science (IISc), Bengaluru

Summer Internship Program 2024

Bharath Udupa, [bharath.u03@gmail.com](mailto:bharath.u03@gmail.com), +91 8431637606

| This program synthesizes a population of 50,000 agents with similar characteristics to the given data

**Step 1: Read the CSV file**

function read\_csv(file\_path):

Initialize an empty list called 'data' to store the rows of data from the CSV file

Open the CSV file specified by 'file\_path' in read mode

Create a CSV reader object to read the file

For each row in the CSV file:

Append the row as a dictionary to the 'data' list

Close the CSV file

Return the 'data' list

**Step 2: Analyze the Sample Data**

function analyze\_data(data):

Initialize an empty dictionary called 'frequency\_counts' to store frequency counts of attribute values

Calculate the total population size by finding the length of the 'data' list

For each row in the 'data' list:

For each key-value pair in the row dictionary:

If the key is not present in 'frequency\_counts':

Initialize an empty dictionary for the key

If the value is not present in the dictionary for the key:

Initialize the count for the value to 0

Increment the count for the value by 1

Initialize an empty dictionary called 'probabilities' to store probabilities of attribute values

For each attribute in 'frequency\_counts':

For each value-count pair in the dictionary for the attribute:

Calculate the probability of the value occurring by dividing the count by the total population size

Add the value-probability pair to the 'probabilities' dictionary under the corresponding attribute

Return the 'probabilities' dictionary

**Step 3: Generate Synthetic Population**

function generate\_population(sample\_data, population\_size):

Initialize an empty list called 'synthetic\_population' to store the synthetic population

Repeat 'population\_size' times:

Initialize an empty dictionary called 'agent' to represent an individual in the synthetic population

For each attribute, probabilities pair in the 'sample\_data' dictionary:

Generate a random choice of attribute value based on the probabilities

Add the chosen attribute value to the 'agent' dictionary

Append the 'agent' dictionary to the 'synthetic\_population' list

Return the 'synthetic\_population' list

**Step 4: Computing Frequencies**

function compute\_frequencies(population):

Initialize an empty dictionary called 'frequencies' to store frequencies of attribute values in the population

For each agent in the 'population' list:

For each attribute-value pair in the agent dictionary:

If the attribute is not present in 'frequencies':

Initialize an empty dictionary for the attribute

If the value is not present in the dictionary for the attribute:

Initialize the count for the value to 0

Increment the count for the value by 1

Return the 'frequencies' dictionary

**Step 5: Write Outputs to Text Files**

function write\_to\_text\_file(frequencies, output\_file):

Open the output text file specified by 'output\_file'in write mode

Create dictionaries to map category numbers to their corresponding category names for each attribute

For each attribute, values pair in the 'frequencies' dictionary:

Write the attribute name followed by a newline character to the file

for each category, count pair in the value\_dict:

convert the category to an integer

if the attribute is 'Sex':

map the category number to the corresponding sex category name using the 'sex\_mapping' dictionary

else if the attribute is 'Age\_category':

map the category number to the corresponding age category name using the 'age\_mapping' dictionary

else:

map the category number to the corresponding education level category name using the 'edu\_mapping' dictionary

write the category name and its count to the file followed by a newline character

write a newline character to separate attributes

close the file

**Main Program**

If the script is executed directly:

Define the file paths for the sample data CSV file and the output text file

Read the CSV file to obtain the sample data

Analyze the sample data to calculate probabilities of attribute values

Generate a synthetic population based on the calculated probabilities

Compute frequencies of attribute values in the synthetic population

Write the computed frequencies to an output txt file