

**National University of Technology**

**Artificial Intelligence Lab-CS3502**

**MSE Submitted to:**

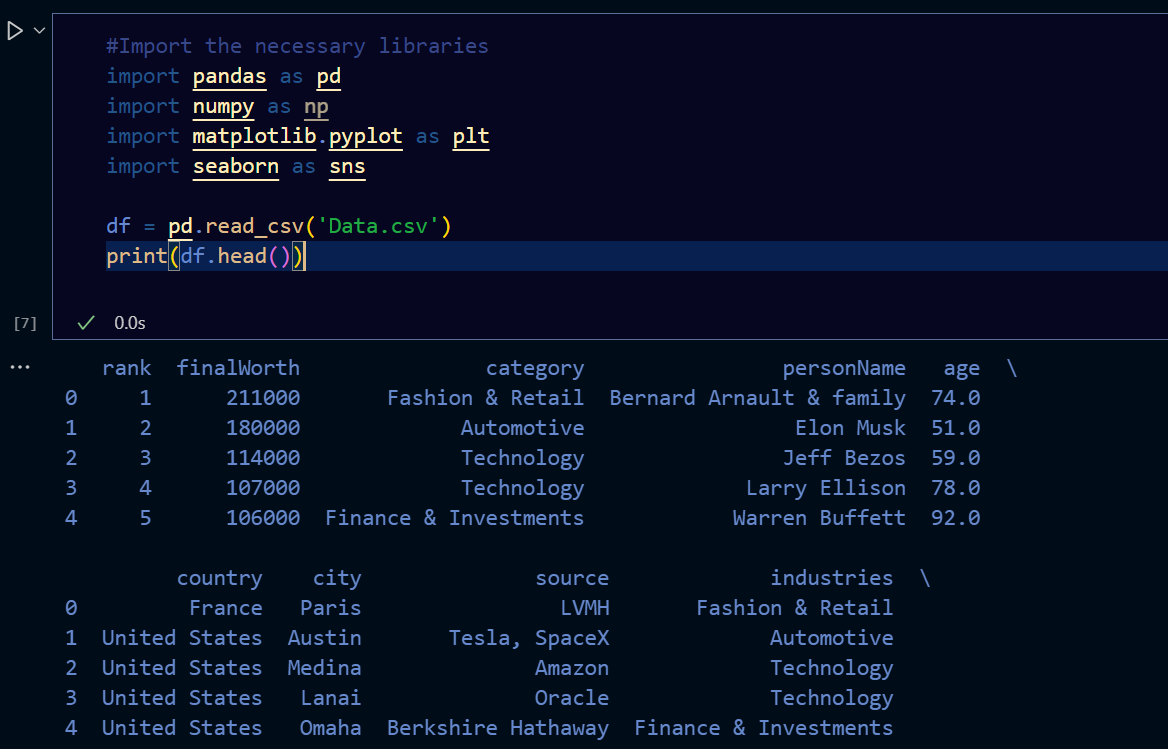
**Sir Yosha Jawad**

**Submitted By:**  
 **Muhammad Abdullah**

Reg. No: **F22607010**

Batch: **Fall AI 2022**

**Question No .1:**

****

**A screenshot of a computer screen

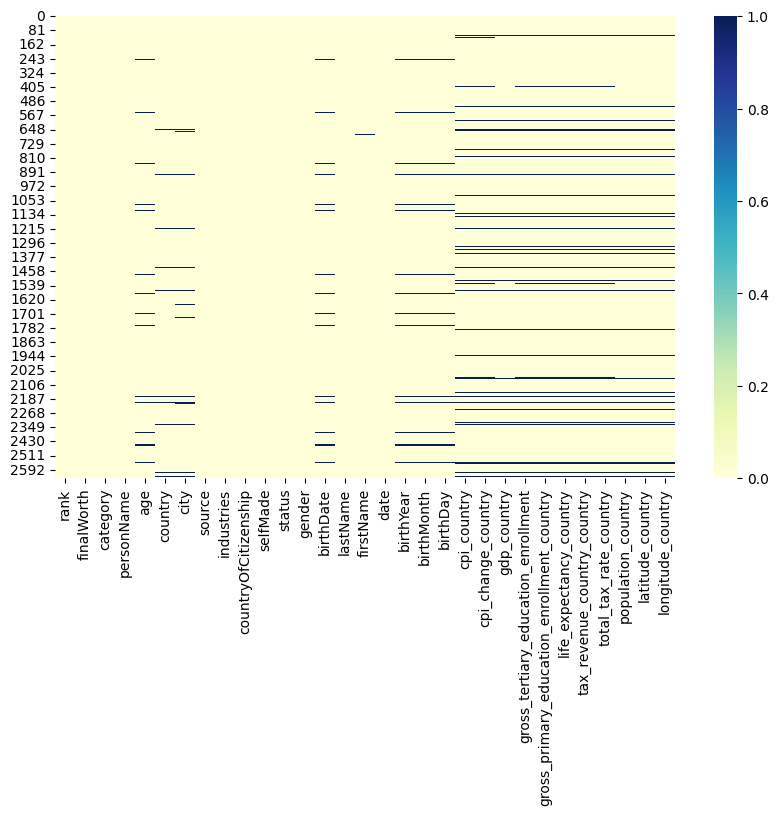
Description automatically generated**

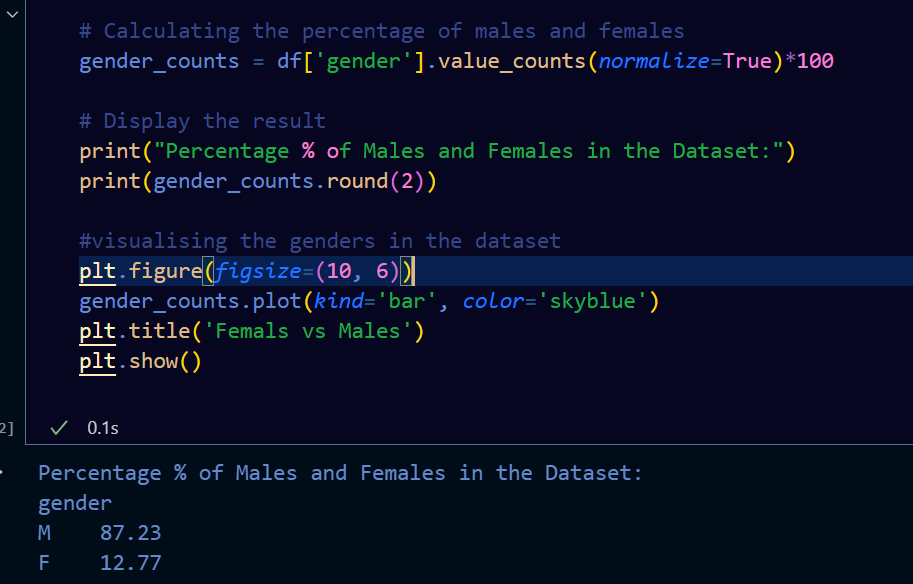
**A screenshot of a computer screen

Description automatically generated**

**A computer screen shot of a program code

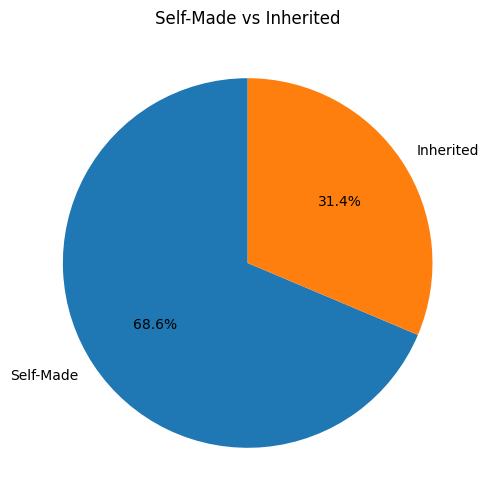
Description automatically generated**

****

****

**A graph with blue rectangles

Description automatically generated**

****

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**A graph of a number of industries

Description automatically generated**

****

**A graph with blue squares

Description automatically generated**

**A computer screen shot of a code

Description automatically generated**

**A graph of blue bars

Description automatically generated**

**A computer screen shot of text

Description automatically generated**

**A computer screen with colorful text

Description automatically generated**

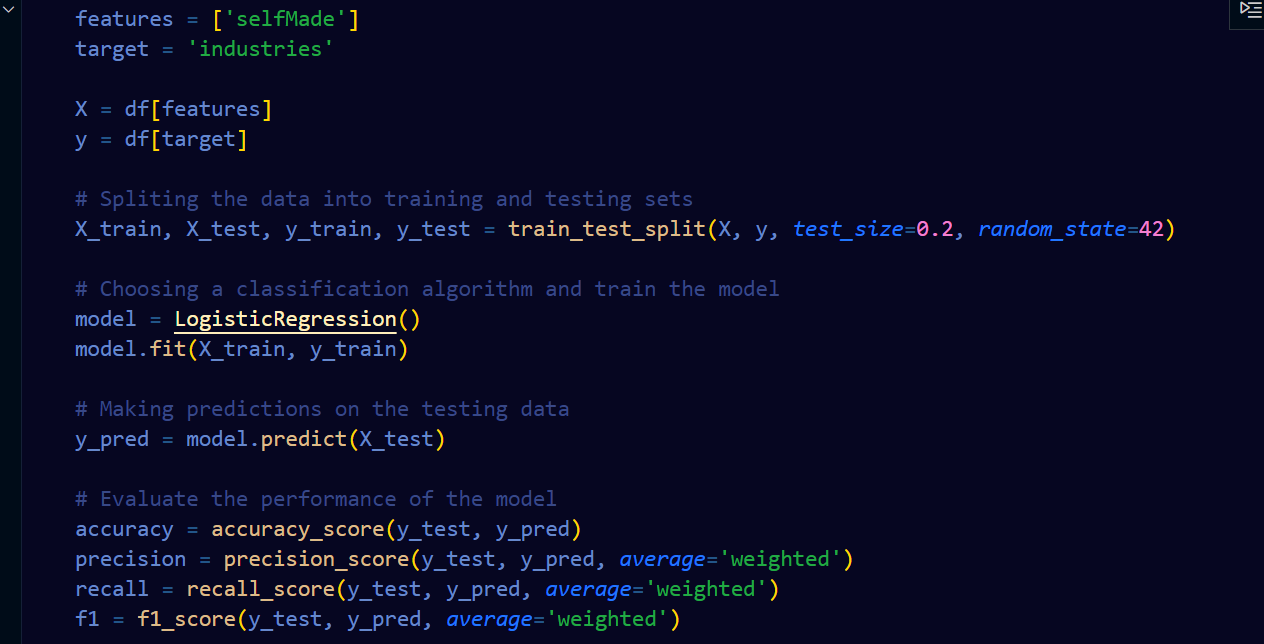
**A screen shot of a computer

Description automatically generated**

****

**A screen shot of a computer code

Description automatically generated**

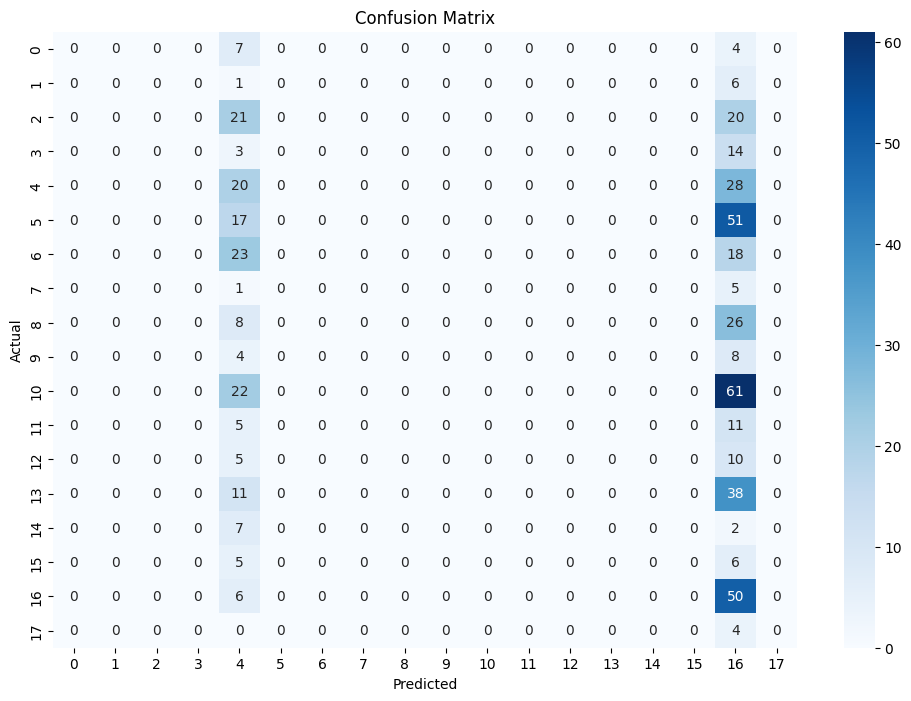
****

**A computer screen with colorful text

Description automatically generated**

**A blue numbers on a black background

Description automatically generated**

****

**Question No .2:**

from collections import deque

*def* bfs(*graph*, *root*):

    visited = set()

    queue = deque([*root*])

    while queue:

        vertex = queue.popleft()

        print(vertex, *end*=" ")

        for neighbour in *graph*[vertex]:

            if neighbour not in visited:

                visited.add(neighbour)

                queue.append(neighbour)

graph = {

    'A': ['B', 'D'],

    'B': ['U', 'L'],

    'D': ['L'],

    'U': [],

    'L': ['A'],

    'H': ['G'],

}

bfs(graph, 'B')

**Output:**

****