Practical No : 1
Name: Thorave Avishkar Shrikrushna

Roll No : 65

Class:BE Artificial Intelligances & Data Sciences

Title : Write a program for pre-processing of a text document such as stop word removal, stemming

```
In [1]: import nltk
        nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Amruta\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
Out[1]: True
```

```
In [3]: from nltk.corpus import stopwords
```

```
In [5]: nltk.download('stopwords')
        print(stopwords.words('english'))
```

```
['a', 'about', 'above', 'after', 'again', 'against', 'ain', 'all', 'am', 'an', 'a
nd', 'any', 'are', 'aren', "aren't", 'as', 'at', 'be', 'because', 'been', 'befor
e', 'being', 'below', 'between', 'both', 'but', 'by', 'can', 'couldn', "could
n't", 'd', 'did', 'didn', "didn't", 'do', 'does', 'doesn', "doesn't", 'doing', 'd
on', "don't", 'down', 'during', 'each', 'few', 'for', 'from', 'further', 'had',
'hadn', "hadn't", 'has', 'hasn', "hasn't", 'have', 'haven', "haven't", 'having',
'he', "he'd", "he'll", 'her', 'here', 'hers', 'herself', "he's", 'him', 'himsel
f', 'his', 'how', 'i', "i'd", 'if', "i'll", "i'm", 'in', 'into', 'is', 'isn', "is
n't", 'it', "it'd", "it'll", "it's", 'its', 'itself', "i've", 'just', 'll', 'm',
'ma', 'me', 'mightn', "mightn't", 'more', 'most', 'mustn', "mustn't", 'my', 'myse
lf', 'needn', "needn't", 'no', 'nor', 'not', 'now', 'o', 'of', 'off', 'on', 'onc
e', 'only', 'or', 'other', 'our', 'ours', 'ourselves', 'out', 'over', 'own', 'r
e', 's', 'same', 'shan', "shan't", 'she', "she'd", "she'll", "she's", 'should',
'shouldn', "shouldn't", "should've", 'so', 'some', 'such', 't', 'than', 'that',
"that'll", 'the', 'their', 'theirs', 'them', 'themselves', 'then', 'there', 'thes
e', 'they', "they'd", "they'll", "they're", "they've", 'this', 'those', 'throug
h', 'to', 'too', 'under', 'until', 'up', 've', 'very', 'was', 'wasn', "wasn't",
'we', "we'd", "we'll", "we're", 'were', 'weren', "weren't", "we've", 'what', 'whe
n', 'where', 'which', 'while', 'who', 'whom', 'why', 'will', 'with', 'won', "wo
n't", 'wouldn', "wouldn't", 'y', 'you', "you'd", "you'll", 'your', "you're", 'you
rs', 'yourself', 'yourselves', "you've"]
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Amruta\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
In [7]: from nltk.corpus import stopwords
        from nltk.tokenize import word_tokenize
```

```
In [31]: example_sent = "my name is amruta just don't forget it"
```

```
In [33]: stop_words = set(stopwords.words('english'))
```

```
In [35]: word_tokens = word_tokenize(example_sent)
```

In [37]:

```
filtered_sentence = [w for w in word_tokens if not w.lower() in stop_words]
```

In [103…

```
filtered_sentence= []
```

In [105…

```
print(w)
```

```
it
```

In [107…

```
for w in word_tokens:
    if w not in stop_words:
        filtered_sentence.append(w)
```

In [109…

```
print(word_tokens)
```

```
['my', 'name', 'is', 'amruta', 'just', 'do', "n't", 'forget', 'it']
```

In [111…

```
print(filtered_sentence)
```

```
['name', 'amruta', "n't", 'forget']
```

In [113…

```
## performing stopwords operation in file
```

In [115…

```
import io
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import os
os.getcwd()
```

Out[115…

```
'C:\\Users\\Amruta'
```

In [117…

```
stop_words = set(stopwords.words('english'))
```

In [119…

```
import os
print("Current working directory:")
print(os.getcwd())
```

```
Current working directory:
C:\Users\Amruta
```

In [121…

```
print("Files in this directory:")
print(os.listdir())
```

```
Files in this directory:
['.anaconda', '.arduinoIDE', '.bash_history', '.conda', '.condarc', '.continuum',
'.eclipse', '.gitconfig', '.ipynb_checkpoints', '.ipython', '.jupyter', '.lesshs
t', '.matplotlib', '.p2', '.streamlit', '.sts4', '.vscode', '.vscode-R', '26-9-25
ds.ipynb', '3D Objects', 'AMRUTA 7 DS.ipynb', 'amruta.py', 'AppData', 'Applicatio
n Data', 'CL-II ( IIOT & IR )', 'Contacts', 'Cookies', 'DATETIME.ipynb', 'Documen
ts', 'Downloads', 'eclipse', 'eclipse-workspace', 'edb_npgsql.exe', 'edb_pgagent_
pg17.exe', 'edb_pgbouncer.exe', 'edb_pgjdbc.exe', 'edb_psqlodbc.exe', 'edb_psqlod
bc.exe-20250922214117', 'edb_psqlodbc.exe-20250922220044', 'edb_psqlodbc.exe-2025
0922220052', 'edb_psqlodbc.exe-20250923215712', 'edb_psqlodbc.exe-2025092321574
2', 'Favorites', 'IntelGraphicsProfiles', 'Links', 'Local Settings', 'marks.csv',
'marks_analysis.py', 'ML Uber 2 Practical .ipynb', 'Music', 'My Documents', 'n=12
345.py', 'NetHood', 'New folder', 'NTUSER.DAT', 'ntuser.dat.LOG1', 'ntuser.dat.LO
G2', 'NTUSER.DAT{bc27eacd-1983-11f0-874b-f0d5bf06ab84}.TM.blf', 'NTUSER.DAT{bc27e
acd-1983-11f0-874b-f0d5bf06ab84}.TMContainer00000000000000000001.regtrans-ms', 'N
TUSER.DAT{bc27eacd-1983-11f0-874b-f0d5bf06ab84}.TMContainer00000000000000000002.r
egtrans-ms', 'ntuser.ini', 'OneDrive', 'pemhttpd.exe', 'postgis_3_5_pg17.exe', 'p
ostgis_3_5_pg17.exe-20250922220100', 'PrintHood', 'Recent', 'Saved Games', 'Searc
hes', 'SendTo', 'Start Menu', 'Templates', 'test', 'uber (1).csv', 'uber.csv', 'U
ntitled.ipynb', 'untitled.txt', 'Untitled1.ipynb', 'Untitled10.ipynb', 'Untitled1
1.ipynb', 'Untitled12.ipynb', 'Untitled13.ipynb', 'Untitled14.ipynb', 'Untitled1
5.ipynb', 'Untitled16.ipynb', 'Untitled17.ipynb', 'Untitled18.ipynb', 'Untitled1
9.ipynb', 'Untitled2.ipynb', 'Untitled20.ipynb', 'Untitled21.ipynb', 'Untitled22.
ipynb', 'Untitled23.ipynb', 'Untitled24.ipynb', 'Untitled25.ipynb', 'Untitled26.i
pynb', 'Untitled27.ipynb', 'Untitled3.ipynb', 'Untitled4.ipynb', 'Untitled5.ipyn
b', 'Untitled6.ipynb', 'Untitled7.ipynb', 'Untitled8.ipynb', 'Untitled9.ipynb',
'Videos', 'Wine.csv']
```

In [132…

```python
file1 = open("untitled.txt", 'r')
print(file1.read())
```

In [134…

```python
line = file1.read()
words = line.split()
```

In [136…

```python
file1 = open("untitled.txt", 'w')
```

In [138…

```python
print(words)
```

```
[]
```

In [140…

```python
for r in words:
    if not r in stop_words:
        appendFile=open('untitled.text','a')
        appendFile.write(" "+r)
        appendFile.close()
```

In [ ]:

```python
# stemming
```

In [142…

```python
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
ps = PorterStemmer()
```

In [144…

```python
words = ["program", "programs", "programmer" ,"progamming" ,"programmers"]
```

In [146…

```python
for w in words:
    print(w,":", ps.stem(w))
```

```
program : program
programs : program
programmer : programm
progamming : progam
programmers : programm
```

In [ ]:

```
#code 2 (stemming words from sentences)
```

In [148…

```
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize
ps = PorterStemmer()
sentence = "Programmer program with programming languages"
words = word_tokenize(sentence)
for w in words :
    print(w,":" , ps.stem(w))
```

```
Programmer : programm
program : program
with : with
programming : program
languages : languag
```

Practical No : 2
Name: Thorave Avishkar Shrikrushna

Roll No: 65

Class: BE AI&DS

Title : Implement a program for retrieval of documents using inverted files.

In [1]:

```
class InvertedIndex:
    def __init__(self):
        # Initialize an empty dictionary to store the inverted index
        self.index = {}

    def add_document(self, doc_id, document):
        # Tokenize the document into words
        words = document.split()
        for word in words:
            word = word.lower()  # Convert to lowercase for case-insensitive sea
            if word not in self.index:
                self.index[word] = []
            if doc_id not in self.index[word]:
                self.index[word].append(doc_id)

    def query(self, words):
        # Tokenize the query into words
        words = words.split()
        results = set()
        for word in words:
            word = word.lower()  # Convert to lowercase for case-insensitive sea
            if word in self.index:
                if not results:
                    results = set(self.index[word])
                else:
                    results &= set(self.index[word])
        return list(results)

    def display_index(self):
        # Display the contents of the inverted index
        for word, doc_ids in self.index.items():
            print(f"{word}: {doc_ids}")


# Example usage
documents = {
    1: "The quick brown fox jumps over the lazy dog",
    2: "Never jump over the lazy dog quickly",
    3: "Brown foxes are quick and jump high"
}

# Create an instance of the InvertedIndex
inverted_index = InvertedIndex()

# Add documents to the index
for doc_id, content in documents.items():
    inverted_index.add_document(doc_id, content)

# Display the inverted index
inverted_index.display_index()
```

```
# Query the index
query = "quick fox"
results = inverted_index.query(query)
print(f"Documents containing '{query}': {results}")
```

```
the: [1, 2]
quick: [1, 3]
brown: [1, 3]
fox: [1]
jumps: [1]
over: [1, 2]
lazy: [1, 2]
dog: [1, 2]
never: [2]
jump: [2, 3]
quickly: [2]
foxes: [3]
are: [3]
and: [3]
high: [3]
Documents containing 'quick fox': [1]
```

3rd

Title: Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using the standard Heart Disease Data Set (You can use Python ML library classes/API.

Name : Thorave Avishkar Shrikrushna

Roll No : 65

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
df=pd.read_csv(r"C:\Users\pansa\Datasets\heart.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

In [4]:

```
df.tail()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

In [6]:

```
df.describe()
```

Out[6]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6

In [7]:

```
df.columns
```

Out[7]:

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [8]:

```
df.isnull().sum()
```

Out[8]:

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [9]:

```
from pgmpy.models import BayesianNetwork
from pgmpy.estimators import MaximumLikelihoodEstimator, HillClimbSearch, BicScore
from pgmpy.inference import VariableElimination
```

In [10]:

```
model = BayesianNetwork([
    ('age', 'trestbps'),
    ('age', 'fbs'),
    ('sex', 'trestbps'),
    ('sex', 'chol'),
    ('trestbps', 'target'),
    ('chol', 'target'),
    ('fbs', 'target')
])
```

In [11]:

```
model.nodes()
```

Out[11]:

```
NodeView(('age', 'trestbps', 'fbs', 'sex', 'chol', 'target'))
```

In [12]:

```
model.edges()
```

Out[12]:

```
OutEdgeView([('age', 'trestbps'), ('age', 'fbs'), ('trestbps', 'target'), ('fbs', 'target'), ('sex', 'trestbps'
), ('sex', 'chol'), ('chol', 'target')])
```

In [13]:

```
model.fit(df, estimator=MaximumLikelihoodEstimator)
```

In [14]:

```
for cpd in model.get_cpds():
    print(cpd)
```

```
+---------+------------+
| age(29) | 0.00390244 |
+---------+------------+
| age(34) | 0.00585366 |
+---------+------------+
| age(35) | 0.0146341  |
+---------+------------+
| age(37) | 0.00585366 |
+---------+------------+
| age(38) | 0.0117073  |
+---------+------------+
| age(39) | 0.0136585  |
+---------+------------+
| age(40) | 0.0107317  |
+---------+------------+
| age(41) | 0.0312195  |
+---------+------------+
| age(42) | 0.0253659  |
+---------+------------+
| age(43) | 0.0253659  |
+---------+------------+
| age(44) | 0.035122   |
+---------+------------+
| age(45) | 0.0243902  |
+---------+------------+
| age(46) | 0.022439   |
+---------+------------+
| age(47) | 0.017561   |
+---------+------------+
| age(48) | 0.022439   |
+---------+-----------+
| age(49) | 0.0165854  |
+---------+-----------+
| age(50) | 0.0204878  |
+---------+-----------+
| age(51) | 0.0380488  |
+---------+-----------+
| age(52) | 0.0419512  |
+---------+-----------+
| age(53) | 0.0253659  |
+---------+-----------+
| age(54) | 0.0517073  |
+---------+-----------+
| age(55) | 0.0292683  |
+---------+-----------+
| age(56) | 0.0380488  |
+---------+-----------+
| age(57) | 0.0556098  |
+---------+-----------+
| age(58) | 0.0663415  |
+---------+-----------+
| age(59) | 0.044878   |
+---------+-----------+
| age(60) | 0.0360976  |
+---------+-----------+
| age(61) | 0.0302439  |
+---------+-----------+
| age(62) | 0.0360976  |
+---------+-----------+
| age(63) | 0.0312195  |
+---------+-----------+
| age(64) | 0.0331707  |
+---------+-----------+
| age(65) | 0.0263415  |
+---------+-----------+
| age(66) | 0.0243902  |
+---------+-----------+
| age(67) | 0.0302439  |
+---------+-----------+
| age(68) | 0.0117073  |
+---------+-----------+
| age(69) | 0.00878049 |
+---------+-----------+
| age(70) | 0.0136585  |
+---------+-----------+
| age(71) | 0.0107317  |
+---------+-----------+
| age(74) | 0.00292683 |
+---------+-----------+
| age(76) | 0.00292683 |
+---------+-----------+
| age(77) | 0.00292683 |
+---------+-----------+
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| age           | age(29)             | age(29) | ... | age(76)             | age(77)             | age(77) |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| sex           | sex(0)              | sex(1)  | ... | sex(1)              | sex(0)              | sex(1)  |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(94)  | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(100) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(101) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(102) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(104) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(105) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(106) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(108) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(110) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(112) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(114) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(115) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(117) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(118) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(120) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(122) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(123) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(124) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(125) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 1.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(126) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(128) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(129) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(130) | 0.02040816326530612 | 1.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(132) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(134) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(135) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(136) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(138) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(140) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(142) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(144) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(145) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(146) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(148) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(150) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(152) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(154) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(155) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(156) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(160) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(164) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(165) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(170) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(172) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(174) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(178) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(180) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(192) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
| trestbps(200) | 0.02040816326530612 | 0.0     | ... | 0.02040816326530612 | 0.02040816326530612 | 0.0     |
+---------------+---------------------+---------+-----+---------------------+---------------------+---------+
+--------+---------+---------+---------+---------+-----+---------------------+---------+---------+---------+
| age    | age(29) | age(34) | age(35) | age(37) | ... | age(71)             | age(74) | age(76) | age(77) |
+--------+---------+---------+---------+---------+-----+---------------------+---------+---------+---------+
| fbs(0) | 1.0     | 1.0     | 1.0     | 1.0     | ... | 0.6363636363636364  | 1.0     | 1.0     | 1.0     |
+--------+---------+---------+---------+---------+-----+---------------------+---------+---------+---------+
| fbs(1) | 0.0     | 0.0     | 0.0     | 0.0     | ... | 0.36363636363636365 | 0.0     | 0.0     | 0.0     |
+--------+---------+---------+---------+---------+-----+---------------------+---------+---------+---------+
+--------+---------+
| sex(0) | 0.30439 |
+--------+---------+
| sex(1) | 0.69561 |
+--------+---------+
+-----------+----------------------+----------------------+
| sex       | sex(0)               | sex(1)               |
+-----------+----------------------+----------------------+
| chol(126) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(131) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(141) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(149) | 0.01282051282051282  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(157) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(160) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(164) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(166) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(167) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(168) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(169) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(172) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(174) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(175) | 0.0                  | 0.015427769985974754 |
+-----------+----------------------+----------------------+
| chol(176) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(177) | 0.009615384615384616 | 0.015427769985974754 |
+-----------+----------------------+----------------------+
| chol(178) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(180) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(182) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(183) | 0.01282051282051282  | 0.0                  |
+-----------+----------------------+----------------------+
| chol(184) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(185) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(186) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(187) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(188) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(192) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(193) | 0.0                  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(195) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(196) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(197) | 0.03205128205128205  | 0.012622720897615708 |
+-----------+----------------------+----------------------+
| chol(198) | 0.009615384615384616 | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(199) | 0.009615384615384616 | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(200) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(201) | 0.009615384615384616 | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(203) | 0.0                  | 0.016830294530154277 |
+-----------+----------------------+----------------------+
| chol(204) | 0.019230769230769232 | 0.021037868162692847 |
+-----------+----------------------+----------------------+
| chol(205) | 0.01282051282051282  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(206) | 0.0                  | 0.011220196353436185 |
+-----------+----------------------+----------------------+
| chol(207) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(208) | 0.0                  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(209) | 0.022435897435897436 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(210) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(211) | 0.009615384615384616 | 0.014025245441795231 |
+-----------+----------------------+----------------------+
| chol(212) | 0.0                  | 0.025245441795231416 |
+-----------+----------------------+----------------------+
| chol(213) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(214) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(215) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(216) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(217) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(218) | 0.0                  | 0.011220196353436185 |
+-----------+----------------------+----------------------+
| chol(219) | 0.009615384615384616 | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(220) | 0.01282051282051282  | 0.011220196353436185 |
+-----------+----------------------+----------------------+
| chol(221) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(222) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(223) | 0.009615384615384616 | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(224) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(225) | 0.02564102564102564  | 0.0                  |
+-----------+----------------------+----------------------+
| chol(226) | 0.009615384615384616 | 0.014025245441795231 |
+-----------+----------------------+----------------------+
| chol(227) | 0.0                  | 0.011220196353436185 |
+-----------+----------------------+----------------------+
| chol(228) | 0.01282051282051282  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(229) | 0.0                  | 0.016830294530154277 |
+-----------+----------------------+----------------------+
| chol(230) | 0.0                  | 0.015427769985974754 |
+-----------+----------------------+----------------------+
| chol(231) | 0.0                  | 0.014025245441795231 |
+-----------+----------------------+----------------------+
| chol(232) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(233) | 0.0                  | 0.016830294530154277 |
+-----------+----------------------+----------------------+
| chol(234) | 0.022435897435897436 | 0.019635343618513323 |
+-----------+----------------------+----------------------+
| chol(235) | 0.0                  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(236) | 0.019230769230769232 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(237) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(239) | 0.009615384615384616 | 0.014025245441795231 |
+-----------+----------------------+----------------------+
| chol(240) | 0.01282051282051282  | 0.014025245441795231 |
+-----------+----------------------+----------------------+
| chol(241) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(242) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(243) | 0.009615384615384616 | 0.014025245441795231 |
+-----------+----------------------+----------------------+
| chol(244) | 0.019230769230769232 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(245) | 0.0                  | 0.012622720897615708 |
+-----------+----------------------+----------------------+
| chol(246) | 0.0                  | 0.014025245441795231 |
+-----------+----------------------+----------------------+
| chol(247) | 0.0                  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(248) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(249) | 0.009615384615384616 | 0.011220196353436185 |
+-----------+----------------------+----------------------+
| chol(250) | 0.009615384615384616 | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(252) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(253) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(254) | 0.009615384615384616 | 0.019635343618513323 |
+-----------+----------------------+----------------------+
| chol(255) | 0.0                  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(256) | 0.009615384615384616 | 0.011220196353436185 |
+-----------+----------------------+----------------------+
| chol(257) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(258) | 0.009615384615384616 | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(259) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(260) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(261) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(262) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(263) | 0.01282051282051282  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(264) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(265) | 0.022435897435897436 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(266) | 0.0                  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(267) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(268) | 0.022435897435897436 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(269) | 0.041666666666666664 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(270) | 0.0                  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(271) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(273) | 0.0                  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(274) | 0.0                  | 0.012622720897615708 |
+-----------+----------------------+----------------------+
| chol(275) | 0.009615384615384616 | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(276) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(277) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(278) | 0.01282051282051282  | 0.0                  |
+-----------+----------------------+----------------------+
| chol(281) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(282) | 0.0                  | 0.019635343618513323 |
+-----------+----------------------+----------------------+
| chol(283) | 0.009615384615384616 | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(284) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(286) | 0.0                  | 0.011220196353436185 |
+-----------+----------------------+----------------------+
| chol(288) | 0.022435897435897436 | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(289) | 0.0                  | 0.011220196353436185 |
+-----------+----------------------+----------------------+
| chol(290) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(293) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(294) | 0.019230769230769232 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(295) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(298) | 0.0                  | 0.008415147265077139 |
+-----------+----------------------+----------------------+
| chol(299) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(300) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(302) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(303) | 0.019230769230769232 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(304) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(305) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(306) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(307) | 0.01282051282051282  | 0.0                  |
+-----------+----------------------+----------------------+
| chol(308) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(309) | 0.0                  | 0.015427769985974754 |
+-----------+----------------------+----------------------+
| chol(311) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(313) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(315) | 0.0                  | 0.009817671809256662 |
+-----------+----------------------+----------------------+
| chol(318) | 0.009615384615384616 | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(319) | 0.01282051282051282  | 0.0                  |
+-----------+----------------------+----------------------+
| chol(321) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(322) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(325) | 0.009615384615384616 | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(326) | 0.0                  | 0.004207573632538569 |
+-----------+----------------------+----------------------+
| chol(327) | 0.01282051282051282  | 0.0                  |
+-----------+----------------------+----------------------+
| chol(330) | 0.01282051282051282  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(335) | 0.0                  | 0.011220196353436185 |
+-----------+----------------------+----------------------+
| chol(340) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(341) | 0.01282051282051282  | 0.0                  |
+-----------+----------------------+----------------------+
| chol(342) | 0.01282051282051282  | 0.0                  |
+-----------+----------------------+----------------------+
| chol(353) | 0.0                  | 0.005610098176718092 |
+-----------+----------------------+----------------------+
| chol(354) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(360) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(394) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(407) | 0.01282051282051282  | 0.0                  |
+-----------+----------------------+----------------------+
| chol(409) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(417) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
| chol(564) | 0.009615384615384616 | 0.0                  |
+-----------+----------------------+----------------------+
+-----------+--------------+---------------+-----+---------------+---------------+---------------+
| chol      | chol(126)    | chol(126)     | ... | chol(564)     | chol(564)     | chol(564)     |
+-----------+--------------+---------------+-----+---------------+---------------+---------------+
| fbs       | fbs(0)       | fbs(0)        | ... | fbs(1)        | fbs(1)        | fbs(1)        |
+-----------+--------------+---------------+-----+---------------+---------------+---------------+
| trestbps  | trestbps(94) | trestbps(100) | ... | trestbps(180) | trestbps(192) | trestbps(200) |
+-----------+--------------+---------------+-----+---------------+---------------+---------------+
| target(0) | 0.5          | 0.5           | ... | 0.5           | 0.5           | 0.5           |
+-----------+--------------+---------------+-----+---------------+---------------+---------------+
| target(1) | 0.5          | 0.5           | ... | 0.5           | 0.5           | 0.5           |
+-----------+--------------+---------------+-----+---------------+---------------+---------------+
```

In [15]:

```
# Perform inference
inference = VariableElimination(model)

# Example query: Probability of having heart disease given specific conditions
query_result = inference.query(variables=['target'], evidence={'age': 55, 'sex': 1, 'trestbps': 140, 'chol': 24

print(query_result)
```

```
+-----------+---------------+
| target    |   phi(target) |
+===========+===============+
| target(0) |        0.5000 |
+-----------+---------------+
| target(1) |        0.5000 |
+-----------+---------------+
```

```
In [ ]:
```

Loading [MathJax]/extensions/Safe.js

CL-II 4 IR

July 22, 2025

[ ]:
```
# Implement Agglomerative hierarchical clustering algorithm using
# appropriate dataset.
```

[ ]:
```
Name : Thorave Avishkar Shrikrushna
Roll No : 65
Course : AI&DS
Class : BE
Sub : CLII
```

[18]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.cluster import AgglomerativeClustering
from sklearn.preprocessing import StandardScaler, normalize
from sklearn.metrics import silhouette_score
import scipy.cluster.hierarchy as shc
```

[34]:
```
#Step 2: Loading and Cleaning the data
df=pd.read_csv("Customer_Data.csv")
```

[36]:
```
X.head()
```

[36]:
```
     ID  Year_Birth   Education Marital_Status    Income  Kidhome  Teenhome \
0  1000        1978         PhD       Together  24342.06        1         0
1  1001        1991      Master          Widow  51784.68        0         0
2  1002        1968       Basic       Divorced  65007.28        2         1
3  1003        1954  Graduation          Widow  52286.31        1         0
4  1004        1982         PhD       Together  40979.04        1         0

  Dt_Customer  Recency  MntWines  …  NumWebVisitsMonth  AcceptedCmp3  \
0  01-01-2012       38       373  ...                  1             1
1  12-01-2012       90        64  ...                  8             1
2  23-01-2012       73       145  ...                  3             0
3  03-02-2012       89       223  ...                  2             1
4  14-02-2012       18       238  ...                  1             1

   AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  \
0          1          0          0          1          1
1          0          0          0          0          1
2          0          1          1          1          0
3          0          0          1          1          1
4          0          0          0          0          0

   Z_CostContact  Z_Revenue  Response
0              3         11         1
1              3         11         0
2              3         11         1
3              3         11         1
4              3         11         1

[5 rows x 25 columns]
```

[38]:

```python
# Handling the missing values
X.ffill(inplace=True)
```

[40]:

```python
X = df.select_dtypes(include=[float, int])
```

[42]:

```python
# Scaling the data so that all the features become comparable
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

[44]:

```python
X.dropna(inplace=True)
```

[46]:

```python
# Normalizing the data so that the data approximately
# follows a Gaussian distribution
X_normalized = normalize(X_scaled)
```

[48]:

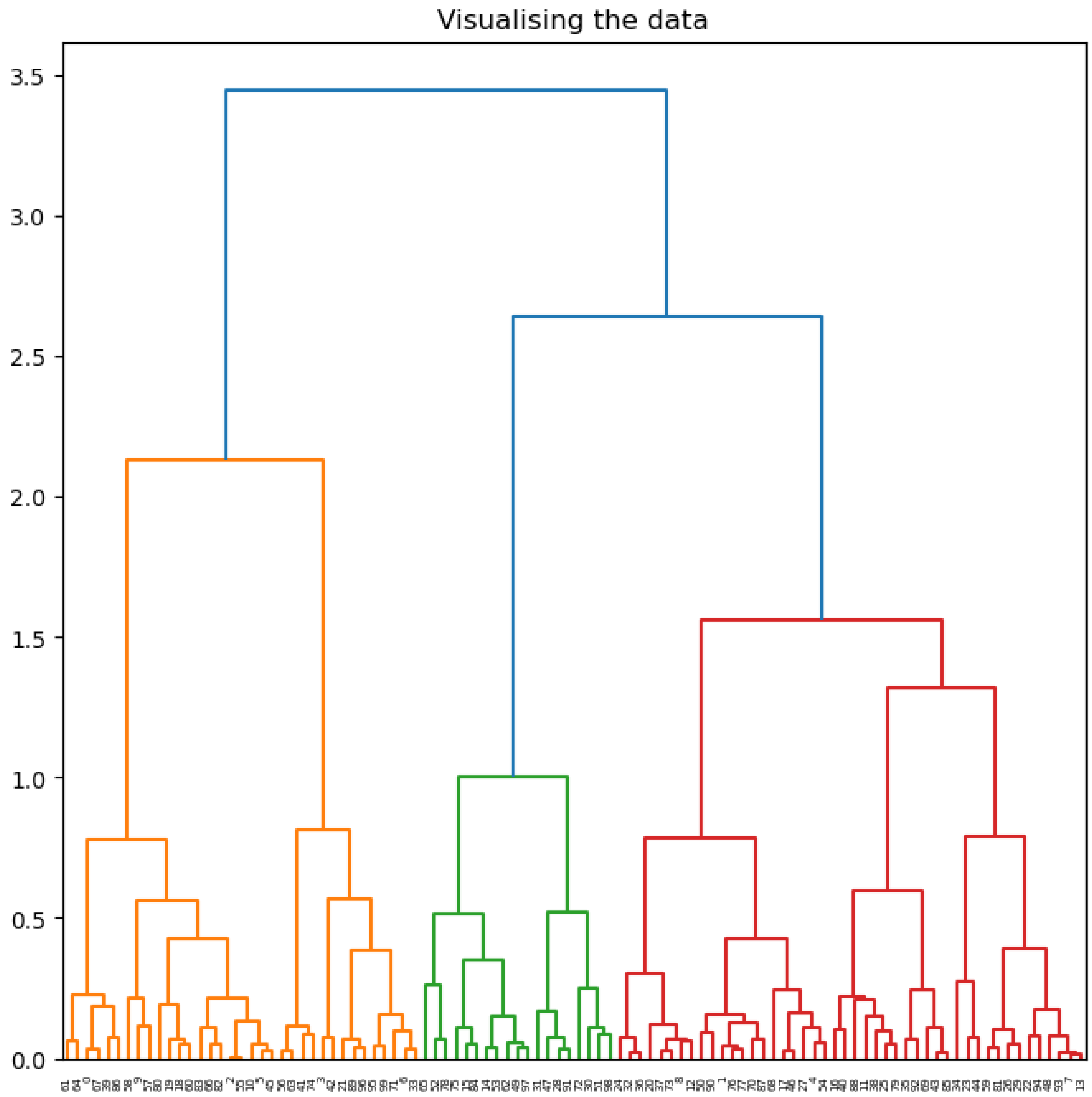
```python
# Converting the numpy array into a pandas DataFrame
X_normalized = pd.DataFrame(X_normalized)
```

[50]:

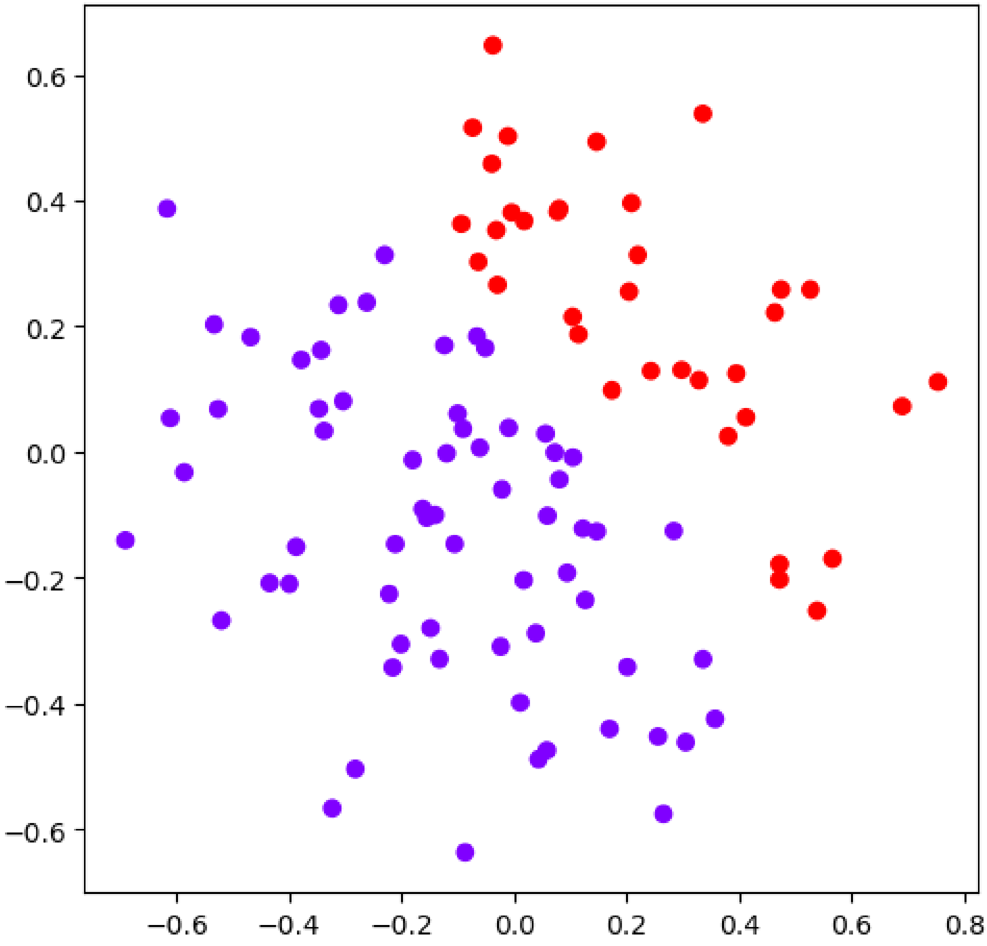
```python
#Step 4: Reducing the dimensionality of the Data
pca = PCA(n_components = 2)
X_principal = pca.fit_transform(X_normalized)
X_principal = pd.DataFrame(X_principal)
X_principal.columns = ['P1', 'P2']
```

[52]:

```python
 #Step 5: Visualizing the working of the Dendrograms
plt.figure(figsize =(8, 8))
plt.title('Visualising the data')
Dendrogram = shc.dendrogram((shc.linkage(X_principal, method ='ward')))
```

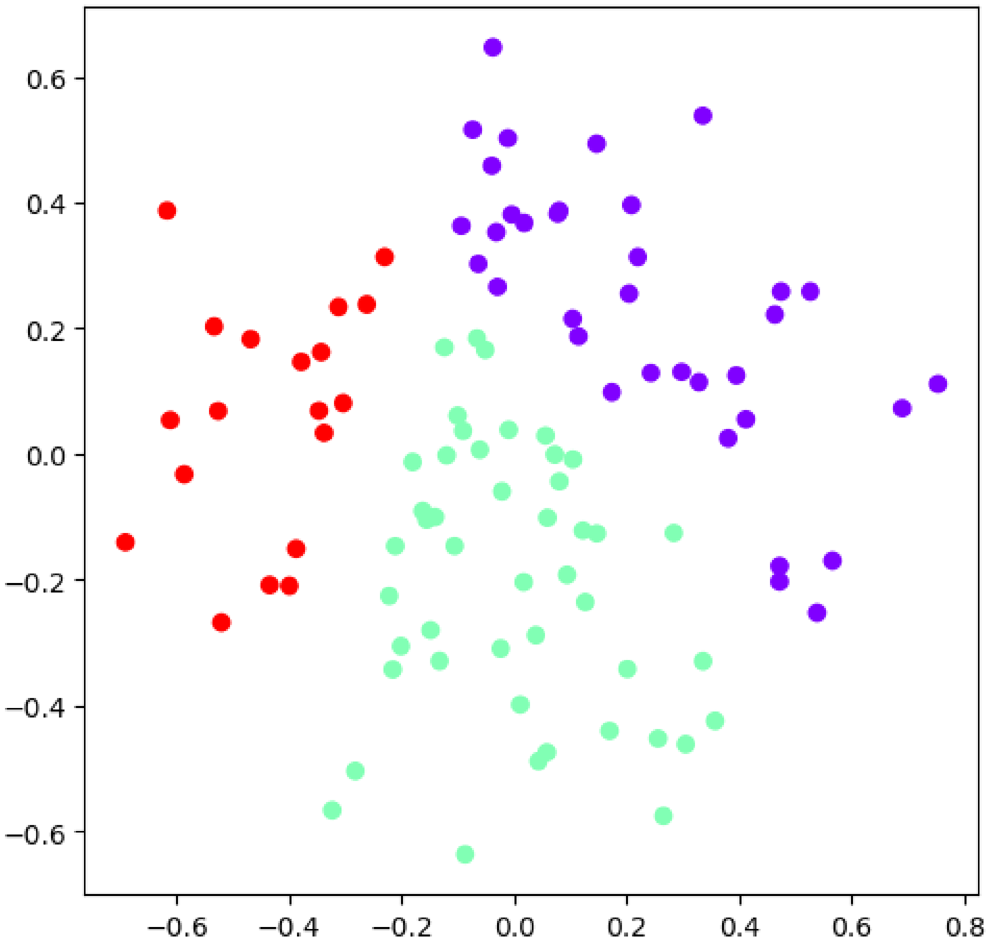


[53]:
```
#Step 6: Building and Visualizing the different clustering models for different␣
↪values of k a) k = 2
ac2 = AgglomerativeClustering(n_clusters = 2)
# Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
c = ac2.fit_predict(X_principal), cmap ='rainbow')
plt.show()
```



[56]:

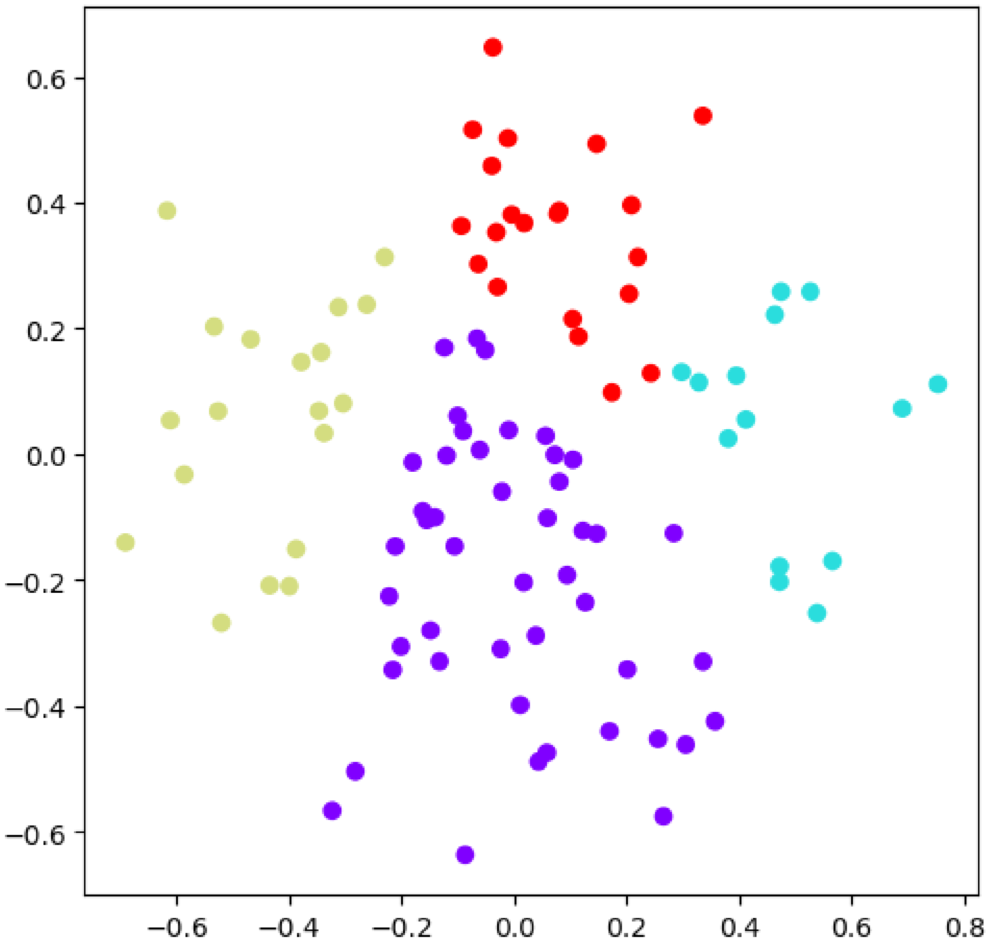
```
ac3 = AgglomerativeClustering(n_clusters = 3)
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'],     X_principal['P2'],
c = ac3.fit_predict(X_principal), cmap ='rainbow')
plt.show()
```



[58]:

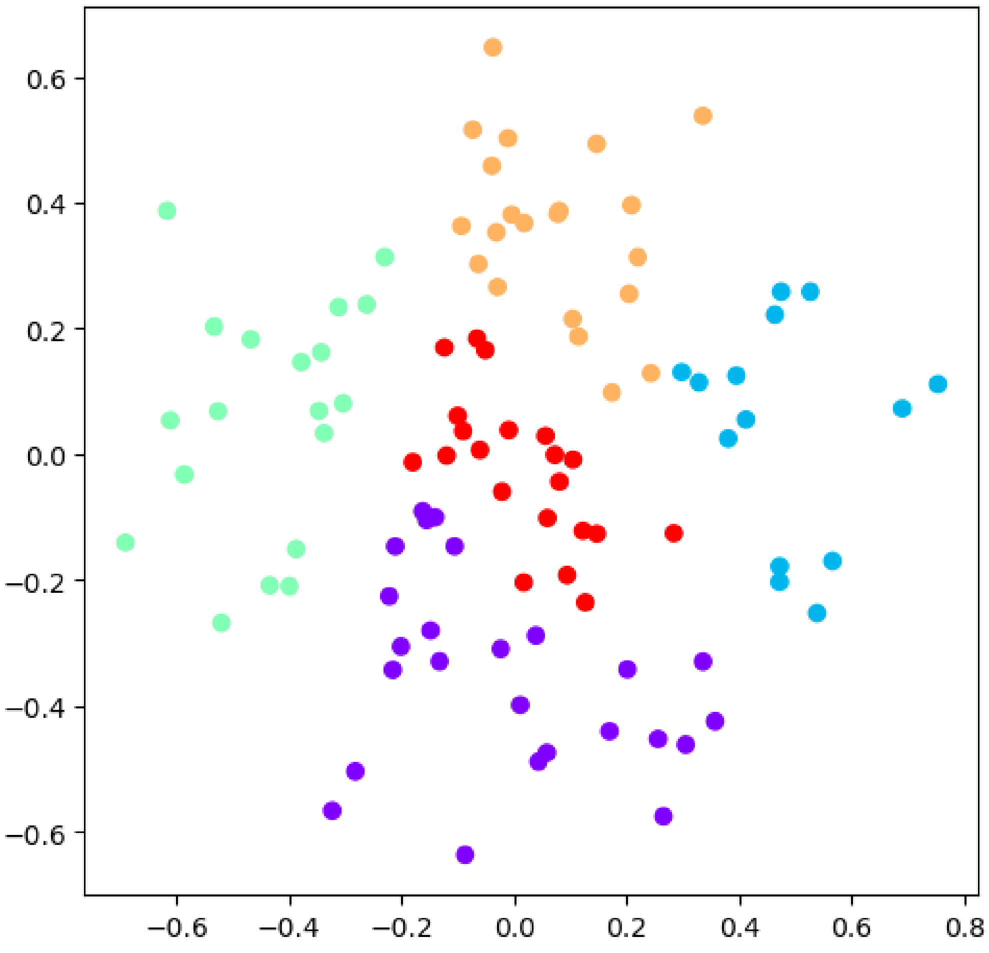
```
ac4 = AgglomerativeClustering(n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'],  X_principal['P2'],
c = ac4.fit_predict(X_principal), cmap ='rainbow')
plt.show()
```



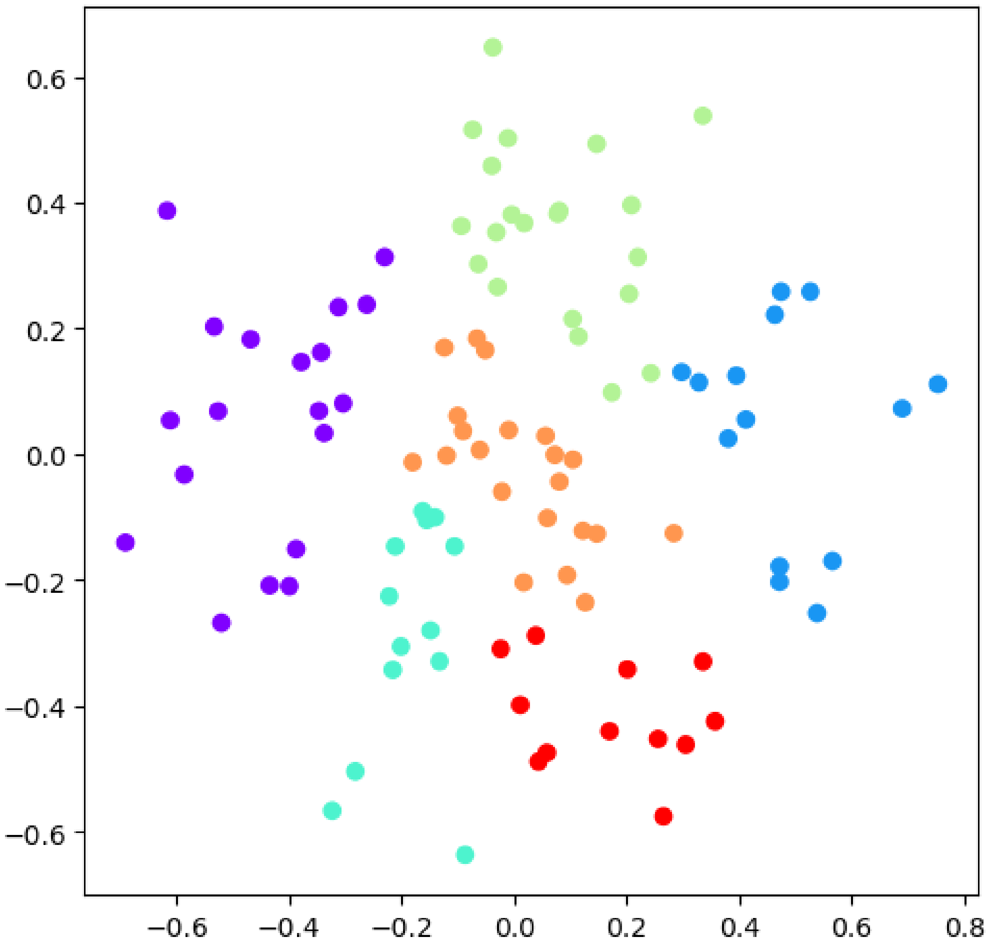
[60]:

```
ac5 = AgglomerativeClustering(n_clusters = 5)
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'], c = ac5.
  sfit_predict(X_principal), cmap ='rainbow')
plt.show()
```



[62]:

```
ac6 = AgglomerativeClustering(n_clusters = 6)
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'],     X_principal['P2'],
c = ac6.fit_predict(X_principal), cmap ='rainbow')
plt.show()
```



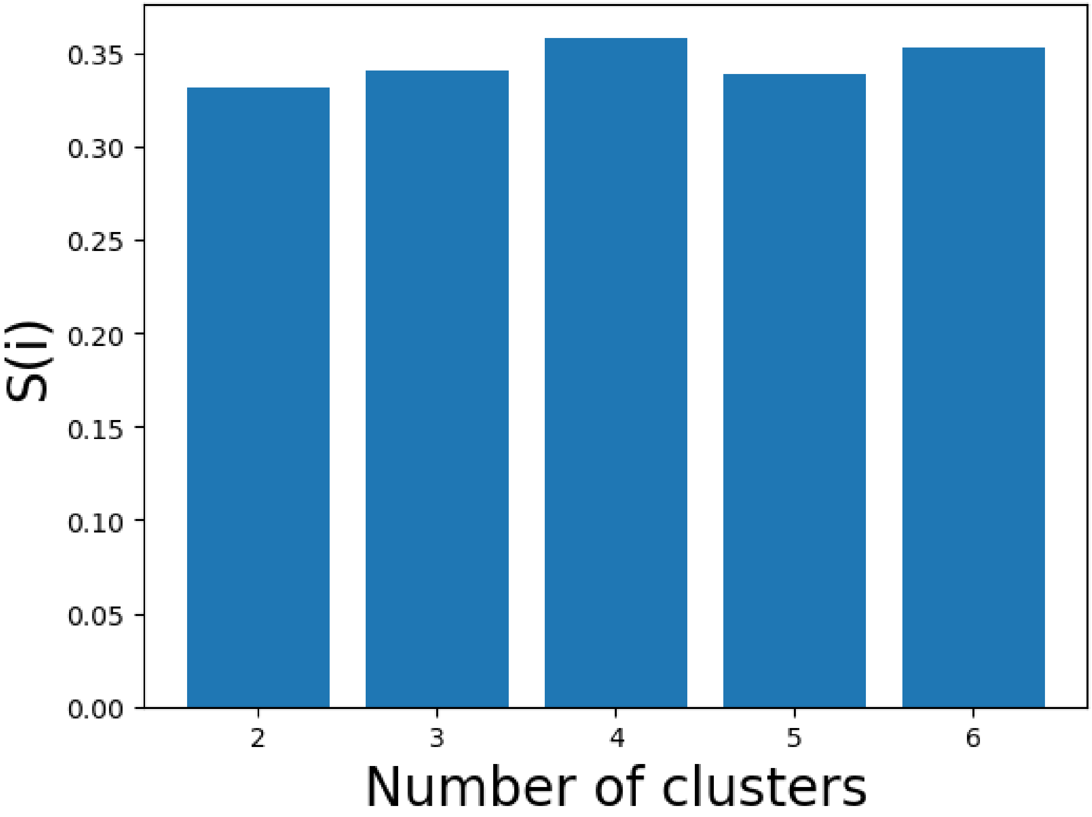
[64]:

```
#Step 7: Evaluating the different models and Visualizing the
#results.
k = [2, 3, 4, 5, 6]

# Appending the silhouette scores of the different models to the list
silhouette_scores = []
silhouette_scores.append(
silhouette_score(X_principal, ac2.fit_predict(X_principal)))
silhouette_scores.append(
silhouette_score(X_principal, ac3.fit_predict(X_principal)))
silhouette_scores.append(
silhouette_score(X_principal, ac4.fit_predict(X_principal)))
silhouette_scores.append(
silhouette_score(X_principal, ac5.fit_predict(X_principal)))
silhouette_scores.append(
silhouette_score(X_principal,   ac6.fit_predict(X_principal)))
```

[66]:

```
# Plotting a bar graph to compare the results
plt.bar(k, silhouette_scores)
plt.xlabel('Number of clusters', fontsize = 20)
plt.ylabel('S(i)', fontsize = 20)
plt.show()
```



[ ]:

5th

Name: Thorave Avishkar Shrikrushna

Roll No: 65

Class: BE AI&DS

Title:Implement PAge Rank Algorithm

Practical No:5

In [1]:

```
import requests
from bs4 import BeautifulSoup
from urllib.parse import urljoin
import numpy as np

# Function to get all the links from a webpage
def get_links(url):
    try:
        response = requests.get(url)
        soup = BeautifulSoup(response.content, 'html.parser')
        links = set()

        for link in soup.find_all('a', href=True):
            absolute_url = urljoin(url, link['href'])
            if absolute_url.startswith('http'):
                links.add(absolute_url)
        return links
    except Exception as e:
        print(f"Error fetching {url}: {e}")
        return set()

# Function to build the link graph
def build_graph(start_url, depth=2):
    pages = {start_url}  # Initialize the set of pages with the start URL
    graph = {}

    # Crawl pages up to the given depth
    for _ in range(depth):
        new_pages = set()
        for page in pages:
            if page not in graph:  # Only process pages that haven't been proces
                links = get_links(page)  # Get links from the current page
                graph[page] = links      # Store the links in the graph
                new_pages.update(links)  # Add newly discovered links to new_pag
        pages.update(new_pages)  # Update pages to include newly found pages

    return graph

# Example: Starting from a single URL
start_url = "https://example.com"
depth = 2  # Define the depth here

link_graph = build_graph(start_url, depth)

# PageRank implementation
def page_rank(graph, iterations=100, d=0.85):
    pages = list(graph.keys())
    n = len(pages)

    # Initialize PageRank values
    ranks = np.ones(n) / n
    # Create adjacency matrix
    adjacency_matrix = np.zeros((n, n))

    for i, page in enumerate(pages):
        for link in graph[page]:
            if link in pages:
                j = pages.index(link)
                adjacency_matrix[j, i] = 1.0 / len(graph[page])

    # PageRank iterative process
    for _ in range(iterations):
        ranks = (1 - d) / n + d * adjacency_matrix.dot(ranks)

    # Mapping pages back to their PageRank values
    page_rank_dict = {pages[i]: ranks[i] for i in range(n)}
    return page_rank_dict

# Compute PageRank
ranks = page_rank(link_graph)
for page, rank in ranks.items():
    print(f"{page}: {rank:.4f}")
```

```
https://example.com: 0.0750
https://www.iana.org/domains/example: 0.1388
```