

# Source code of PowerMethod

26 de noviembre de 2020

## 1. Matrix.py

---

```
1 import numpy as np
2 import csv
3 from Util import Util
4
5 class Matrix:
6
7     def __init__(self, *args):
8
9         # A new matrix is created by giving the args[0] parameter
10        # if args[0] is an integer then it will create a square
11        # matrix with this parameter.
12        if isinstance(args[0], int):
13            self.size = args[0]
14            self.matrix = np.transpose(np.random.dirichlet(np.ones(args[0]), size=args[0]))
15            self.rows, self.columns = self.matrix.shape
16
17        # if args[0] is a string (the path of csv file) then it will load the
18        # csv file in a new matrix
19        elif isinstance(args[0], str):
20            util = Util
21            self.file = args[0]
22            file = open(self.file)
23            rows = len(file.readlines())
24            file.close()
25            self.rows = rows
26            with open(self.file, 'r') as nFile:
27                line = nFile.readline()
28                self.columns = line.count(',') + 1
29                self.matrix = np.empty((self.rows, self.columns))
30
31            #fill the matrix
32            with open(args[0]) as csv_file:
33                csv_reader = csv.reader(csv_file, delimiter=',')
34                f = 0
35                for row in csv_reader:
36                    for c in range(self.columns):
37                        self.matrix[f, c] = util.stringToFloat(row[c])
38                    f += 1
39
40            # creates a canonical vector
41            self.canonicalVector = np.empty((self.rows, 1))
42            for f in range(self.rows):
43                if (f == 0):
44                    self.canonicalVector[f, 0] = 1
45                else:
46                    self.canonicalVector[f, 0] = 0
47
48     def getNumberOfRows(self):
49         return self.rows
50
51     def getNumberOfColumns(self):
52         return self.columns
53
54     def getMatrix(self):
55         return self.matrix
56
57     def getCanonicalVector(self):
58         return self.canonicalVector;
59
60     def setCanonicalVector(self, canonicalVector):
61         self.canonicalVector = canonicalVector
```

---

## 2. Operation.py

---

```
1 from Matrix import Matrix
2 import numpy as np
3
4 class Operation:
5
6     def __init__(self, iterationNumber, arg):
7         self.iterationNumber = iterationNumber
8         self.arg = arg
9         self.iteration = []
10
11     def powerMethod(self):
12         self.iteration = []
13         self.result = None
14         matrix = Matrix(self.arg)
15         value=self.checkStochastic(matrix)
16
17         if value == True:
18             for x in range(self.iterationNumber):
19                 self.result=np.dot(matrix.getMatrix(),matrix.getCanonicalVector())
20                 matrix.setCanonicalVector(self.result)
21                 self.iteration.append(np.transpose(self.result))
22             return True
23         else:
24             return False
25
26     def getIteration(self):
27         return self.iteration
28
29     def checkStochastic(self,matrix):
30         listOfValues = np.sum(matrix.getMatrix(),axis=0)
31         for x in listOfValues:
32             if round(x,4) != 1:
33                 return False
34         return True
```

---