Report and screenshots of Datamining advanced project

CS 634, spring 2019

Programming language: python 3.7

SID: 31470206

Program

In this project I used anaconda spyder IDE to implement HITS algorithm for some search terms. The main function of this program looks like this:

```
if __name__ == "__main__":

    k = int(input("define Upper bound for added pages (k): " or 5 )) # uper bound of ad
ded pages for each seed page
    N = int(input("define Upper bound for printed pages (N): " or 5 )) # uper bound of
added pages for each seed page
    wholeSet, adj = buildWholeSet()
    hits(nodes, adj, error_rate)
    printPages(nodes,N,k)
    if printGraph:
        showGraph(nodes, adj)
```

There are two user inputs as the upper bound for number of added pages for each page in the root set (each seed) which is denoted by k, and there is another bound N for the number of pages printed in the output results based on the order of authority and hub values. As seen in the main function, first the neighborhood graph is built (which is called wholeSet) and then the hits algorithm is run on it and at the end, the results are printed as described in the project descriptions. The general parameters of the program can be seen here:

```
'''''
##################################################
set parameters
##################################################
'''

searchQuery = "deep learning"
header = {'user-agent': 'Mozilla/5.0'}

rootSetFile = 'G:/My Drive/Sem2/DataMining/Projects/advanced/RootSet.txt'
baseSet1File = 'G:/My Drive/Sem2/DataMining/Projects/advanced/baseSet1.txt'
baseSet2File = 'G:/My Drive/Sem2/DataMining/Projects/advanced/baseSet2.txt'
allPages = 'G:/My Drive/Sem2/DataMining/Projects/advanced/allPgaes.txt'
adjfile = 'G:/My Drive/Sem2/DataMining/Projects/advanced/adjfile.txt'

online_search = 1 # 1 to search online, 0 to use file from memory
use_google_package = 1 # 1 to use google search library, 0 to use requests
saveToFile = 1
printNeighbourhood = 1 # 0 to avoid printing all pages in neighbourhood graph
printGraph = 1
seed = 30 # number of seed pages
error_rate = 0.001
```

In case one wants to save the files and the outputs, they can set the `saveToFile` flag to 1. Also there is an option to search for the results online or just read the data from files. There is another option to use googlesearch library for building the root set in the beginning of the program or else one can use requests package to search for the results on google. Below the important functions of the program are explained.

There are three global variables that are used throughout the program:

```
addedPages = [0]*seed # for the upper-bound k
nodes = []
adj = []
```

Each page is a node and has an ID, an address, authority and hub values.

```
class node:
    def __init__(self, pageid, url, auth, hub):
        self.pageid = pageid
        self.url = url
        self.auth = auth
        self.hub = hub
```

Each page is a node and has an ID, an address, authority and hub values. There are many links that are not acceptable as a valid url address, so they should be filtered using the function below which checks for links to remove unacceptable ones.

```
def validLink(url):
    wrong = ['facebook', 'twitter', 'linkedin', 'youtube', 'deeplearning4j', 'slideshare', 'doubleclick', 'ads', '.png', '.jpg', '.svg', '.png']
    validLink = bool(re.match(r'^(?:https?:\/\/)(?:[\w-]+)(?:\.[\w-]+)*\.[\w]+(?:\/[^\n]+)?$', url))
    invalidLink = bool(re.match(r'^.*\.(jpg|JPG|gif|GIF|doc|DOC|pdf|PDF)$',url))

    if (
            validLink == True and
            invalidLink == False and
            all(url.find(t)==-1 for t in wrong)
        ):
        return True
    else:
        return False
```

After getting the inputs from the user the program tries to build the root set first. In `build_root_set` function, the search query is used to find results and put them in nodes.

Then the program starts to find the linked pages and the linking pages in turn to complete the neighborhood graph using two functions, `addLinkedPages` and `addLinkingPages`. Then the function `print_desccriptions1` is used to print the final pages in a format similar to Google's.

For the HITS part, in the function hits, the values of the authority and hub are stored in different arrays. At each step, first the value of authorities is updated, based on the sum of the hub values of the incoming links. Then they are normalized using norm variable. After all the authority values are updated and normalized, we do the same process for the hub values. Using converge function, we check whether all the updated values are different from the previous values by a margin larger than the error rate. If yes, we continue the updated. If not, we stop the process of hits and try to print the resulting pages in order of their authority and hub values. The convergence function looks like:

```
def converge(nodes, errorrate, auth, auth_prev, hub, hub_prev):
    counter = 0;
    for i in range(len(nodes)):
        if ((abs(auth[i] - auth_prev[i]) < errorrate) and (abs(hub[i] - hub_prev[i]) < errorrate)):
            counter = counter + 1;
```

```
    if counter == len(nodes): #converge if all differences are less than errorrate
        return 1
    else:
        return 0
```

This function checks every authority and hub value with their previous values. If all of the differences are less than the error rate threshold, the program returns 1 which means converged.

In the hits function we have:

```
def hits(nodes, adj, errorrate):
    converged = 0
    auth = []
    auth_prev = []
    hub = []
    hub_prev = []

    for p in nodes:
        p.auth = 1
        p.hub = 1
        auth.append(p.auth)
        hub.append(p.hub)
        auth_prev.append(p.auth)
        hub_prev.append(p.hub)

    while True:

        auth_prev = auth[:]
        hub_prev = hub[:]

        #update authority values first
        norm = 0
        for p in nodes:
            #look in the adj list for incoming links to p
            incoming = []
            for i in range(len(adj)):
                if p.pageid in adj[i]:
                    incoming.append(i)

            for q in nodes:    # for q in p.incoming
                if q.pageid in incoming:
                    p.auth = p.auth + q.hub
            norm = norm + math.pow(p.auth,2) #calculate the sum of the squared auth val
ues to normalise
        norm = math.sqrt(norm)

        for p in nodes:
            p.auth = p.auth / norm if norm else 1 #update the auth scores with normaliz
ation

        # update hub values
        norm = 0
        for p in nodes:
            for q in nodes: # for q in p.outgoing
                if q.pageid in adj[p.pageid]:
                    p.hub = p.hub + q.auth
            norm = norm + math.pow(p.hub,2) #calculate the sum of the squared hub value
s to normalise
        norm = math.sqrt(norm)
```

```
47.         for p in nodes:
48.             p.hub = p.hub / norm if norm else 1 #update the hub scores with normalizati
    on
49.             #hub_prev[count] = p.hub
50.
51.         for i in range(len(nodes)):
52.             auth[i] = nodes[i].auth
53.             hub[i] = nodes[i].hub
54.
55.         converged = converge(nodes, errorrate, auth, auth_prev, hub, hub_prev)
56.         if (converged): break
```

first we take a copy of the authority and hub arrays to save them for convergence comparisons. Then the incoming links to a page in adjacency list are defined in a loop to sum up their hub values. Then the resulting authority value is normalized using the sum of squared values. Then the same process is done for hub values, but here we first find the pages that a page is linking to, and then sum up their authority values.

Outputs

In this project, sometimes the linking pages to a seed page cause errors, so I didn't collect those pages for all the results. First all the pages in the neighborhood graph are printed, and then the first N pages in order of their authority and then the first N pages in order of their hub values.

Outputs for term : deep learning

I used an error rate of 0.001 for convergence. The input values are set to k = 20 and N = 5. First the pages in the neighborhood graph are printed (which are 417 pages). In this list there are pages with different languages, and many of them have some problem. For example some of the pages are removed from the web, many of them don't have titles or descriptions, and some pages are blocking the program from accessing. Since the printing of all pages is time consuming, we can set the flag printNeighbourhood to zero:

```
==========================================
    pages in the neighbourhood graph
==========================================
403 Forbidden
 None
 ((authority= 0.000000  |||   hub= 0.000000 ))
------------------------------------------------------------
Deep learning - Wikipedia
 Deep learning  (also known as deep structured learning  or
on the layers used in artificial neural networks. Learning c

 ((authority= 0.000000  |||   hub= 0.005190 ))
------------------------------------------------------------
Category:Deep learning - Wikipedia
 The following 40 pages are in this category, out of  40 tot

 ((authority= 0.000000  |||   hub= 0.011555 ))
------------------------------------------------------------
Deep belief network - Wikipedia
 In machine learning, a deep belief network (DBN) is a gener
multiple layers of latent variables ("hidden units"), with c

 ((authority= 0.000000  |||   hub= 0.011724 ))
------------------------------------------------------------
Semi-supervised learning - Wikipedia
 Semi-supervised learning is a class of machine learning tas
small amount of labeled data with a large amount of unlabele
```

Then the first N=5 pages are printed based on authority and hub values:

```
==============================================
    pages sorted by authority
==============================================
Coursera Help Center
 Official Coursera Help Center. Find answers to your
 ((authority= 0.267159  |||   hub= 0.000000 ))
--------------------------------------------------
Coursera: Top online courses
 Learn on the go with Coursera for iOS. Access free a
others.

App features:
• Access our full catalog: Choose from 2,600+ courses
• Take control of your downtime: Stream v…
 ((authority= 0.267159  |||   hub= 0.000000 ))
--------------------------------------------------
Coursera: Online courses - Apps on Google Play
 Learn anywhere with Coursera. Access more than 2,000
career by mastering subjects from Python programming

Learn from top instructors in an engaging learning ex
```

And then the first N=5 pages based on their hub values:

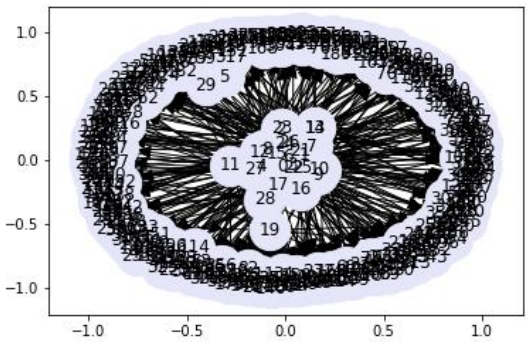
```
==============================================
    pages sorted by hub
==============================================
Deep Learning | Coursera
 Learn Deep Learning from deeplearning.ai. If you want to b
skills in tech. We will help you become good at Deep Learni
 ((authority= 0.000000  |||   hub= 0.706848 ))
------------------------------------------------------------
Neural Networks and Deep Learning | Coursera
 Learn Neural Networks and Deep Learning from deeplearning.
highly sought after, and mastering deep learning will give
 ((authority= 0.000000  |||   hub= 0.706848 ))
------------------------------------------------------------
Semi-supervised learning - Wikipedia
 Semi-supervised learning is a class of machine learning ta
data with a large amount of unlabeled data.  Semi-supervise
(with completely labeled training data).  Many machine-lear
can produce considerable improvement in learning accuracy c
learning (where all data is labeled).[1] The acquisition of
segment) or a physical experiment (e.g. determining the 3D
the labeling process thus may render a fully labeled traini
supervised learning can be of great practical value.  Semi-

 ((authority= 0.000000  |||   hub= 0.020851 ))
------------------------------------------------------------
Deep belief network - Wikipedia
 In machine learning, a deep belief network (DBN) is a gene
variables ("hidden units"), with connections between the la

 ((authority= 0.000000  |||   hub= 0.011724 ))
------------------------------------------------------------
Category:Deep learning - Wikipedia
 The following 40 pages are in this category, out of  40 to

 ((authority= 0.000000  |||   hub= 0.011555 ))
------------------------------------------------------------
```

This is how the graph would look like:



Outputs for term : machine learning

I used an error rate of 0.0001 for convergence. The input values are set to k = 50 and N = 10. The neighborhood graph is 534 pages.

==========

pages sorted by authority

==========

Timeline of machine learning - Wikipedia

This page is a timeline of machine learning. Major discoveries, achievements, milestones and other major events are included.

((authority= 0.139920 ||| hub= 0.003994))

@vas3k • Instagram photos and videos

2,137 Followers, 118 Following, 195 Posts - See Instagram photos and videos from @vas3k

((authority= 0.139920 ||| hub= 0.000000))

Машинное обучение для людей

Разбираемся простыми словами

((authority= 0.139920 ||| hub= 0.000000))

Naive Bayes classifier - Wikipedia

In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

((authority= 0.139920 ||| hub= 0.000000))

Decision tree learning - Wikipedia

In computer science, Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

((authority= 0.139920 ||| hub= 0.000000))

Logistic regression - Wikipedia

In statistics, the logistic model (or logit model) is a widely used statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail, win/lose, alive/dead or healthy/sick; these are represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each dependent variable having its own parameter; for a binary independent variable this generalizes the odds ratio.

((authority= 0.139920 ||| hub= 0.000000))

---

k-nearest neighbors algorithm - Wikipedia

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression.[1] In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

((authority= 0.139920 ||| hub= 0.000000))

Support-vector machine - Wikipedia

In machine learning, support-vector machines (SVMs, also support-vector networks[1]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

((authority= 0.139920 ||| hub= 0.000000))

---

6 Easy Steps to Learn Naive Bayes Algorithm (with code in Python)

This article describes the basic principle behind Naive Bayes algorithm, its application, pros & cons, along with its implementation in Python and R

((authority= 0.139920 ||| hub= 0.000000))

Bayesian poisoning - Wikipedia

Bayesian poisoning is a technique used by e-mail spammers to attempt to degrade the effectiveness of spam filters that rely on Bayesian spam filtering. Bayesian filtering relies on Bayesian probability to determine whether an incoming mail is spam or is not spam. The spammer hopes that the addition of random (or even carefully selected) words that are unlikely to appear in a spam message will cause the spam filter to believe the message to be legitimate—a statistical type II error.

((authority= 0.139920 ||| hub= 0.000000))

==========

pages sorted by hub

==========

Machine Learning for Everyone

In simple words. With real-world examples. Yes, again

((authority= 0.000000 ||| hub= 0.999230))

Machine learning - Wikipedia

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.[1][2]:2 Machine learning algorithms are used in a wide variety of applications, such as email filtering, and computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning.[3][4] In its application across business problems, machine learning is also referred to as predictive analytics.

((authority= 0.000000 ||| hub= 0.038094))

Outline of machine learning - Wikipedia

The following outline is provided as an overview of and topical guide to machine learning. Machine learning is a subfield of soft computing within computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence.[1] In 1959, Arthur Samuel defined machine learning as a "field of study that gives computers the ability to learn without being explicitly programmed".[2] Machine learning explores the study and construction of algorithms that can learn from and make predictions on data.[3] Such algorithms operate by building a model from an example training set of input observations in order to make data-driven predictions or decisions expressed as outputs, rather than following strictly static program instructions.

((authority= 0.000000 ||| hub= 0.007635))

Timeline of machine learning - Wikipedia

This page is a timeline of machine learning. Major discoveries, achievements, milestones and other major events are included.

((authority= 0.139920 ||| hub= 0.003994))

Portal:Machine learning - Wikipedia

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering, and computer vision, where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

((authority= 0.000000 ||| hub= 0.003640))

---

403 Forbidden

None

((authority= 0.000000 ||| hub= 0.001034))

What is machine learning (ML)? - Definition from WhatIs.com

This definition explains the meaning of machine learning and how enterprises are using it to make their applications -- from CRM platforms to operational systems -- smarter.

((authority= 0.000000 ||| hub= 0.000000))

Machine Learning Applications and Platform | SAP Leonardo

Build an intelligent enterprise with machine learning software â“ uniting human expertise and computer insights to improve processes, innovation, and growth.

((authority= 0.000000 ||| hub= 0.000000))

What is machine learning?

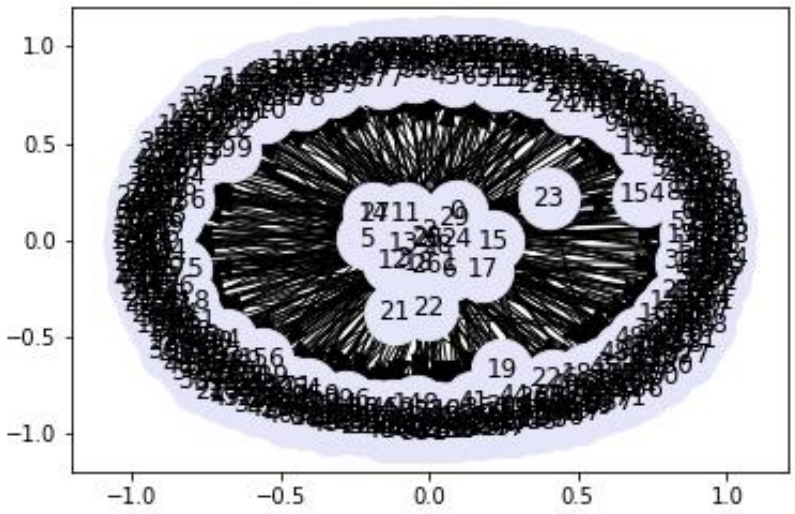
Machine learning algorithms can now approach or exceed human intelligence across a remarkable number of tasks.

((authority= 0.000000 ||| hub= 0.000000))

A Beginner’s Guide to AI/ML 🤔

The ultimate guide to machine learning. Simple, plain-English explanations accompanied by math, code, and real-world examples.

((authority= 0.000000 ||| hub= 0.000000))



Outputs for term : Data Mining

I used an error rate of 0.0001 for convergence. The input values are set to k = 20 and N = 20. The neighborhood graph is 362 pages.

==========

pages sorted by authority

==========

Make your donation now - Wikimedia Foundation

((authority= 0.318941 ||| hub= 0.000000))

Category:Data mining - Wikimedia Commons

wikipedia:vi:Thể loại:Khai thác dữ liệu

((authority= 0.179225 ||| hub= 0.000000))

Category:Data mining

Wikimedia category

((authority= 0.179225 ||| hub= 0.000000))

تصنيف:تنقيب البيانات - ويكيبيديا، الموسوعة الحرة

يشتمل هذا التصنيف على 3 تصنيفات فرعية، من أصل 3.

((authority= 0.179225 ||| hub= 0.000000))

বিষয়শ্রেণী:ডাটা মাইনিং - উইকিপিডিয়া

এই বিষয়শ্রেণীতে কেবল নিচের পাতাটি আছে।

((authority= 0.179225 ||| hub= 0.000000))

Kategorie:Data-Mining – Wikipedia

Themen die zu dem Informatik-Bereich Data-Mining gehören, auch bekannt als Wissensentdeckung in Datenbanken (KDD). Themen in denen lediglich Data-Mining verwendet wird, sollten in diese Kategorie nicht eingetragen werden. Stattdessen kann beispielsweise die Kategorie Kategorie:Business Intelligence passender sein.

((authority= 0.179225 ||| hub= 0.000000))

Κατηγορία:Εξόρυξη δεδομένων - Βικιπαίδεια

Αυτή η κατηγορία περιέχει τις ακόλουθες 8 σελίδες, από 8 συνολικά.

((authority= 0.179225 ||| hub= 0.000000))

Categoría:Minería de datos - Wikipedia, la enciclopedia libre

Esta categoría contiene las siguientes 33 páginas:

((authority= 0.179225 ||| hub= 0.000000))

Kategoria:Datu-meatzaritza - Wikipedia, entziklopedia askea.

Jarraian ageri diren 8 orriak kategoria honetan daude. Kategoria honetan, guztira, 8 orri daude.

((authority= 0.179225 ||| hub= 0.000000))

رده:داده‌کاوی - ویکی‌پدیا، دانشنامهٔ آزاد

مقالهٔ اصلی در این رده عبارت‌است از: داده‌کاوی

((authority= 0.179225 ||| hub= 0.000000))

Catégorie:Exploration de données — Wikipédia

L’exploration de données, aussi connue sous les noms fouille de données, data mining (forage de données) ou encore Extraction de Connaissances (ECD en français, KDD en Anglais), a pour objet l’extraction d'un savoir ou d'une connaissance à partir de grandes quantités de données, par des méthodes automatiques ou semi-automatiques, et l'utilisation industrielle ou opérationnelle de ce savoir.

((authority= 0.179225 ||| hub= 0.000000))

분류:데이터 마이닝 - 위키백과, 우리 모두의 백과사전

다음은 이 분류에 속하는 문서 9개 가운데 9개입니다.

((authority= 0.179225 ||| hub= 0.000000))

Categoria:Data mining - Wikipedia

Questa categoria riguarda il data mining, estrazione di un sapere o di una conoscenza a partire da grandi quantità di dati.

((authority= 0.179225 ||| hub= 0.000000))

קטגוריה:כריית מידע – ויקיפדיה

דף קטגוריה זה כולל את 11 הדפים הבאים, מתוך 11 בקטגוריה כולה. (לתצוגת עץ)

((authority= 0.179225 ||| hub= 0.000000))

Ангилал:Data mining — Википедиа нэвтэрхий толь

Мөн ангид дараах хуудас хамаарна.

((authority= 0.179225 ||| hub= 0.000000))

Category:データマイニング - Wikipedia

このカテゴリには以下の下位カテゴリのみが含まれています。

((authority= 0.179225 ||| hub= 0.000000))

Categoria:Mineração de dados – Wikipédia, a enciclopédia livre

Esta categoria contém as seguintes 2 subcategorias (de um total de 2).

((authority= 0.179225 ||| hub= 0.000000))

Категория:Глубокий анализ данных — Википедия

Эта категория содержит только следующую подкатегорию.

((authority= 0.179225 ||| hub= 0.000000))

Kategória:Data mining – Wikipédia

V tejto kategórii sa nachádzajú 2 stránky z 2 celkom.

((authority= 0.179225 ||| hub= 0.000000))

Категорија:Експлорација података — Википедија, слободна енциклопедија

^ | 0—9 | A | B | C | Č | Ć | D | DŽ | Đ | E | F | G | H | I | J | K | L | LJ | M | N | NJ | O | P | R | S | Š | T | U | V | Z | Ž

((authority= 0.179225 ||| hub= 0.000000))

==========

pages sorted by hub

==========

Category:Data mining - Wikipedia

Data mining facilities are included in some of the Category:Data analysis software and Category:Statistical software products.

((authority= 0.000000 ||| hub= 0.836418))

Examples of data mining - Wikipedia

Data mining, the process of discovering patterns in large data sets, has been used in many applications.

((authority= 0.000000 ||| hub= 0.535559))

Java Data Mining - Wikipedia

Java Data Mining (JDM) is a standard Java API for developing data mining applications and tools. JDM defines an object model and Java API for data mining objects and processes. JDM enables applications to integrate data mining technology for developing predictive analytics applications and tools. The JDM 1.0 standard was developed under the Java Community Process as JSR 73.[1] In 2006, the JDM 2.0 specification was being developed under JSR 247, but has been withdrawn in 2011 without standardization.[2]

((authority= 0.000000 ||| hub= 0.116472))

Data mining - Wikipedia

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems.[1] Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information (with intelligent methods) from a data set and transform the information into a comprehensible structure for further use.[1][2][3][4] Data mining is the analysis step of the "knowledge discovery in databases" process, or KDD.[5] Aside from the raw analysis step, it also involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating.[1] The difference between data analysis and data mining is that data analysis is used to test models and hypotheses on the dataset, e.g., analyzing the effectiveness of a marketing campaign, regardless of the amount of data; in contrast,

data mining uses machine-learning and statistical models to uncover clandestine or hidden patterns in a large volume of data.[6]

((authority= 0.000000 ||| hub= 0.001592))

Educational data mining - Wikipedia

Educational data mining (EDM) describes a research field concerned with the application of data mining, machine learning and statistics to information generated from educational settings (e.g., universities and intelligent tutoring systems). At a high level, the field seeks to develop and improve methods for exploring this data, which often has multiple levels of meaningful hierarchy, in order to discover new insights about how people learn in the context of such settings.[1] In doing so, EDM has contributed to theories of learning investigated by researchers in educational psychology and the learning sciences.[2] The field is closely tied to that of learning analytics, and the two have been compared and contrasted.[3]

((authority= 0.000000 ||| hub= 0.001592))

What is data mining?

Learn how data mining uses machine learning, statistics and artificial intelligence to look for same patterns across a large universe of data.

((authority= 0.000000 ||| hub= 0.001592))

What is data mining? - Definition from WhatIs.com

This definition explains the meaning of data mining and how enterprises can use it to sort through information to make better business decisions.

((authority= 0.000000 ||| hub= 0.001592))

Definition of Data Mining | What is Data Mining ? Data Mining Meaning - The Economic Times

Data Mining definition - What is meant by the term Data Mining ? meaning of IPO, Definition of Data Mining on The Economic Times.

((authority= 0.000000 ||| hub= 0.001592))

Data Mining

((authority= 0.000000 ||| hub= 0.001592))

Data Mining - GeeksforGeeks

In general terms, “Mining” is the process of extraction of some valuable material from the earth e.g. coal mining, diamond mining etc. In the context… Read More »

((authority= 0.000000 ||| hub= 0.000000))

Data Mining: How Companies Use Data to Find Useful Patterns and Trends

Data mining is a process used by companies to turn raw data into useful information by using software to look for patterns in large batches of data.

((authority= 0.000000 ||| hub= 0.000000))

Data Mining | Coursera

Learn Data Mining from University of Illinois at Urbana-Champaign. The Data Mining Specialization teaches data mining techniques for both structured data which conform to a clearly defined schema, and unstructured data which exist in the form of ...

((authority= 0.000000 ||| hub= 0.000000))

Data Mining and Applications Graduate Certificate | Stanford Center for Professional Development

Graduate certificate introduces important new ideas in data mining and machine learning and describes their applications to business, science, and technology

((authority= 0.000000 ||| hub= 0.000000))

Definition of DATA MINING

the practice of searching through large amounts of computerized data to find useful patterns or trends… See the full definition

((authority= 0.000000 ||| hub= 0.000000))

Data Mining Concepts

None

((authority= 0.000000 ||| hub= 0.000000))

Data Mining information, news, and how-to advice

Data Mining | News, how-tos, features, reviews, and videos

((authority= 0.000000 ||| hub= 0.000000))

Data Mining Explained

Data mining is everywhere. Learn what it is, how it's used, benefits, and current trends. This article will also cover leading data mining tools and common questions.

((authority= 0.000000 ||| hub= 0.000000))

The Definitive Guide to Data Mining

What is Data Mining? What are the most effective data mining techniques? Check out our guide.

((authority= 0.000000 ||| hub= 0.000000))

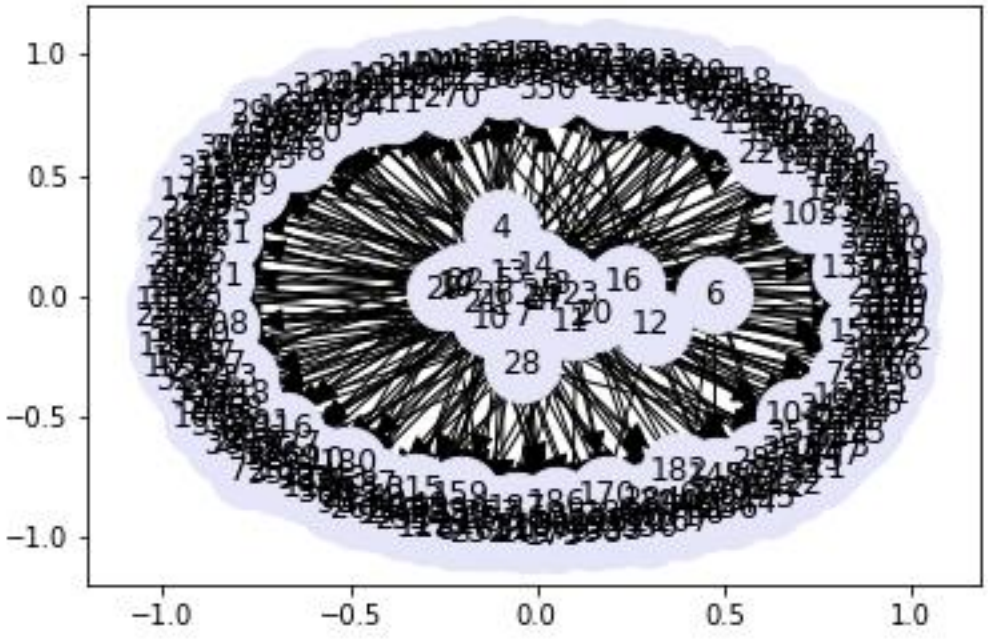
What is Data Mining, Predictive Analytics, Big Data

((authority= 0.000000 ||| hub= 0.000000))

An Introduction to Data Mining - The Data Mining Blog

In this blog post, I will introduce the topic of data mining. The goal is to give a general overview of what is data mining. What is data mining? Data mining is a field of research that has emerged in … Continue reading →

((authority= 0.000000 ||| hub= 0.000000))


1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
4
16
6
12
28

Outputs for term : Big Data Analytics

I used an error rate of 0.0001 for convergence. The input values are set to k = 30 and N = 15. the neighborhood graph is 417 pages.

==========

pages sorted by authority

==========

Login/Register

You forgot to provide an Email Address.

((authority= 0.091287 ||| hub= 0.000000))

Register

You are about to join the ranks of several hundred thousand IT professionals that benefit from the PRO+ premium membership. Members have exclusive and unlimited access to premium content on TechTarget's network of over 120 IT focused websites as well as receive personal invitations to networking events. All you need to do to join is fill out the membership form below.

((authority= 0.091287 ||| hub= 0.000000))

TechTarget - Global Network of Information Technology Websites and Contributors

TechTarget Application Development websites cover application development and architecture, ALM, software testing and QA, BPM, web services, agile, on-premise and cloud development tools and processes, and application project management.

((authority= 0.091287 ||| hub= 0.000000))

Buyer's Guides - Resource from WhatIs.com

Communications platform as a service (CPaaS) is a cloud-based delivery model that allows organizations to add real-time communication capabilities, such as voice, video and messaging, to business applications by deploying application program interfaces (APIs).

((authority= 0.091287 ||| hub= 0.000000))

Q&A: Dinsmore sees open source Apache Spark moving to new stage

Contradictory information on open source Apache Spark performance is giving way to more reasonable assessments, according to analytics expert Thomas Dinsmore.

((authority= 0.091287 ||| hub= 0.000000))

Hortonworks Hadoop distribution goes to two release tracks

Hadoop distribution vendor Hortonworks plans yearly updates for core Hadoop parts, while 'extended services,' such as Hive and Spark, will get more frequent updates.

((authority= 0.091287 ||| hub= 0.000000))

Apache Spark architecture speeds data jobs, ousts MapReduce

At a recent conference, adopters of the Apache Spark architecture highlighted uses of the fast-rising open source standard.

((authority= 0.091287 ||| hub= 0.000000))

Spark Streaming update to address growing torrent of big data

Spark Streaming is taking on greater importance, as big data moves toward real-time performance. Version 2.0 updates are meant to address the issue.

((authority= 0.091287 ||| hub= 0.000000))

NoSQL revs up to the tune of the Spark connector

NoSQL data is getting the analytical treatment using a Spark connector. Riak and Cassandra are just two examples.

((authority= 0.091287 ||| hub= 0.000000))

Co-creator Cutting assesses Hadoop future, present and past

Doug Cutting, co-creator of the Hadoop big data processing framework, discusses the technology's current standing and looks toward the Hadoop future.

((authority= 0.091287 ||| hub= 0.000000))

What is Apache Hadoop YARN? - Definition from WhatIs.com

This definition explains the meaning of Apache Hadoop YARN and how the cluster resource management and job scheduling technology has expanded the types of applications that can be run in Hadoop systems beyond MapReduce batch jobs.

((authority= 0.091287 ||| hub= 0.000000))

What is Apache Spark? - Definition from WhatIs.com

This definition explains Apache Spark, which is an open source parallel process computational framework primarily used for data engineering and analytics.

((authority= 0.091287 ||| hub= 0.000000))

What is big data? - Definition from WhatIs.com

Learn the characteristics of big data, how big data is being used by organizations today, the benefits of big data analytics and the overall challenges around bad data, data privacy and misuse of data.

((authority= 0.091287 ||| hub= 0.000000))

What is big data management? - Definition from WhatIs.com

Big data management is the organization, administration and governance of large volumes of both structured and unstructured data in order to ensure a high level of data quality and accessibility for business purposes, including business intelligence and big data analytics applications.

((authority= 0.091287 ||| hub= 0.000000))

What is data scientist? - Definition from WhatIs.com

This definition explains what a data scientist is, including essential job skills, education requirements and responsibilities.

((authority= 0.091287 ||| hub= 0.000000))

==========

pages sorted by hub

==========

Big Data Analytics - What it is and why it matters

None

((authority= 0.000088 ||| hub= 0.500000))

What is big data analytics? - Definition from WhatIs.com

This definition explains the meaning of big data analytics and how it can help organizations to increase revenues and improve business operations. Learn more about how big data analytics works and the importance it can have for the businesses that use it.

((authority= 0.000000 ||| hub= 0.500000))

Big data - Wikipedia

"Big data" is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software. Data with many cases (rows) offer greater statistical power, while data with higher complexity (more attributes or columns) may lead to a higher false discovery rate.[2] Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating, information privacy and data source. Big data was originally associated with three key concepts: volume, variety, and velocity.[3] Other concepts later attributed with big data are veracity (i.e., how much noise is in the data) [4] and value.[5]

((authority= 0.000000 ||| hub= 0.500000))

Big Data Analytics News, Analysis, & Advice - InformationWeek

InformationWeek shares news, analysis and advice on the tools and strategies that connect the dots across data. Connect with our big data analytics experts.

((authority= 0.000000 ||| hub= 0.500000))

What is Big Data Analytics

Big data analytics enables companies to increase revenues, decrease costs and become more competitive within their industries.

((authority= 0.000000 ||| hub= 0.000371))

Big Data Analytics - Databricks

ALL0-9ABCDEFGHIJKLMNOPQRSTUVWXYZ« Back to Glossary IndexSource DatabricksBefore Hadoop, both storage and compute technology was limited; as a result, the analytics process was long and rigid. In order to get every new data source ready to be stored it had to go through a lengthy process, usually known as ETL. Once the data was ready, it had …

((authority= 0.000000 ||| hub= 0.000001))

Big data analytics solutions: machine data can reveal customer behavior, security threats | Splunk

Splunk Enterprise is the leading platform to collect, analyze and deliver real-time insights from machine-generated big data. Try Splunk Enterprise and Hunk| Splunk Analytics for Hadoop for free.

((authority= 0.000088 ||| hub= 0.000000))

---

403 Forbidden

None

((authority= 0.000000 ||| hub= 0.000000))

Big Data Analytics Courses | Coursera

Learn online and earn valuable credentials from top universities like Yale, Michigan, Stanford, and leading companies like Google and IBM. Join Coursera for free and transform your career with degrees, certificates, Specializations, & MOOCs in data science, computer science, business, and dozens of other topics.

((authority= 0.000000 ||| hub= 0.000000))

Big Data Analytics Software Solution | Alteryx

Turn your databases into Big Data analytics powerhouses with Alteryx in-database analytics. Our platform combines Big Data environments with external datasets to find the maximum value from your data while utilizing every ounce of your database’s native intelligence for analytics. Learn more here!

((authority= 0.000000 ||| hub= 0.000000))

---

403 Forbidden

None

((authority= 0.000000 ||| hub= 0.000000))

Data Science vs. Big Data vs. Data Analytics

This article talks about what is Big Data, Data Analytics, and Data Science and the major differences between the three terminologies.

((authority= 0.000000 ||| hub= 0.000000))

Data and Analytics are your most powerful Asset - Big Data | Teradata

Data and Analytics are your most powerful Asset - In today's high-stakes business environment, leading companies—enterprises that differentiate, outperform, and adapt to customer needs faster than competitors—rely on Big Data Analytics. They see how the purposeful, systematic exploitation of big data, coupled with analytics, reveals opportunities for better business outcomes.

((authority= 0.000088 ||| hub= 0.000000))

What is Big Data Analytics: Definition | Informatica US

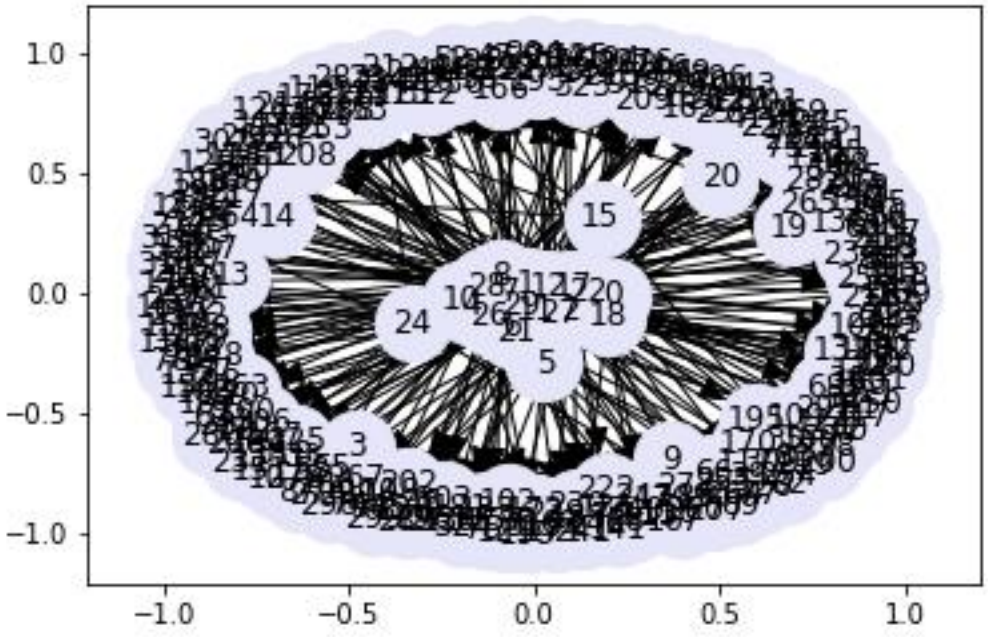
The definition of big data analytics can be found in our guide to data integration technology nomenclature. Discover today & find solutions for tomorrow.

((authority= 0.000000 ||| hub= 0.000000))

Big Data Analytics | Information Builders

Big data analytics and information management help organizations enhance the value of information, through smarter storage, integration,…

((authority= 0.000000 ||| hub= 0.000000))


1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
20
15
24
5
18
3
9
19

Source Code

```
# -*- coding: utf-8 -*-
"""
Created on Tue Apr 16 11:23:24 2019

@author: Hadi
"""
#from googlesearch import search
try:
    import urllib.request as urllib2
except ImportError:
#except Exception:
    import urllib2
from googlesearch import search
from bs4 import BeautifulSoup
import urllib.parse
import io, re, math
import requests
from requests.exceptions import *
#from .excepts import *
#from .helpers import process_image_url
from webpreview import web_preview
import networkx as nx
import matplotlib.pyplot as plt


'''''
##################################################
set parameters
##################################################
'''

searchQuery = "data mining"
header = {'user-agent': 'Mozilla/5.0'}

rootSetFile = 'G:/My Drive/Sem2/DataMining/Projects/advanced/RootSet.txt'
baseSet1File = 'G:/My Drive/Sem2/DataMining/Projects/advanced/baseSet1.txt'
baseSet2File = 'G:/My Drive/Sem2/DataMining/Projects/advanced/baseSet2.txt'
allPages = 'G:/My Drive/Sem2/DataMining/Projects/advanced/allPgaes.txt'
adjfile = 'G:/My Drive/Sem2/DataMining/Projects/advanced/adjfile.txt'

online_search = 1 # 1 to search online, 0 to use file from memory
use_google_package = 1 # 1 to use google search library, 0 to use requests
saveToFile = 1
printNeighbourhood = 0 # 0 to avoid printing all pages in neighbourhood graph
printGraph = 1
seed = 30 # number of seed pages
error_rate = 0.0001

'''''
##################################################
general variables and functions
##################################################
'''
```

```
55.
56. addedPages = [0]*seed # for the upper-bound k
57. nodes = []
58. adj = []
59.
60. class URLopener(urllib2.FancyURLopener):
61.     version = "Mozilla/5.0"
62.     def http_error_default(self, url, fp, errcode, errmsg, headers):
63.         if errcode == 403:
64.             raise ValueError("403")
65.         return super(URLopener, self).http_error_default(
66.             url, fp, errcode, errmsg, headers
67.         )
68.
69. class node:
70.     def __init__(self, pageid, url, auth, hub):
71.         self.pageid = pageid
72.         self.url = url
73.         self.auth = auth
74.         self.hub = hub
75.
76. def searchGoogle(query=None, times=0):
77.     search_results = search(query,tld="com", num=times, stop=times, pause=2);
78.     return search_results
79.
80. def searchYahoo(query=None, times=0):
81.     query = re.sub(r"\s+", '+', query)
82.     yahoo = "https://search.yahoo.com/search?q=" + query + "&n=" + str(times)
83.     result=[]
84.     #opener = URLopener()
85.     try:
86.         #page = opener.open(urllib.parse.unquote(yahoo))
87.         page = urllib2.urlopen(yahoo)
88.         soup = BeautifulSoup(page, "lxml")
89.         #soup = BeautifulSoup(page.read(), features='lxml')
90.         #links = soup.find_all('a', href=True);
91.         #for link in links:
92.         for link in soup.find_all(attrs={"class": "ac-algo"}):
93.         #for link in soup.select("algo"):
94.             #if link.get('class') and "ac-algo" in link['class']:
95.             result.append(link.get('href'))
96.     except (HTTPError, ValueError):
97.     #except Exception:
98.         pass
99.     return(result)
100.
101.        def save_file(filename, savedlist):
102.            with io.open(filename, "w", encoding="utf-8") as f:
103.                    for item in savedlist:
104.                        f.write("%s\n" % item)
105.
106.        def validLink(url):
107.            wrong = ['facebook', 'twitter', 'linkedin', 'youtube', 'deeplearning4j', 'sl
    ideshare', 'doubleclick', 'ads', '.png', '.jpg', '.svg', '.png']
108.            validLink = bool(re.match(r'^(?:https?:\/\/)(?:[\w-]+)(?:\.[\w-
    ]+)*\.[\w]+(?:\/[^\n]+)?$', url))
109.            invalidLink = bool(re.match(r'^.*\.(jpg|JPG|gif|GIF|doc|DOC|pdf|PDF)$',url))
110.
111.            if (
112.                    validLink == True and
```

```
                    invalidLink == False and
                    all(url.find(t)==-1 for t in wrong)
                ):
                    return True
            else:
                return False

        def getDomain(url):
            domain = url.split("//")[-1].split("/")[0]
            return domain


        def getLinks(url):
            #page = opener.open(urllib.parse.unquote(i))
            page = urllib2.urlopen(url)
            soup = BeautifulSoup(page.read(), features='lxml')
            links = soup.findAll("a", href=True)
            #check for ads
            adLinks = soup.findAll("h3", {"class": "sA5rQ"})
            return links


        def checkforAd (url):
            #check for ads
            query1 = query.replace(" ","+")
            url = "https://www.google.com/search?q="+query1
            page = urllib2.urlopen(url)
            soup = BeautifulSoup(page.read(), features='lxml')
            mydivs = soup.findAll("div", {"class": "sA5rQ"})

        '''''
        ##################################################
        build root set
        ##################################################
        '''

        def build_root_set(query):
            RootSet = []
            count = 0

            if use_google_package:

                search_results = searchGoogle(query, seed+10) #Getting 40 results if any duplicates
                for page in search_results:
                    if not page in RootSet and count < seed and validLink(page):
                        RootSet.append(page)
                        node1 = node(count,page,1,1)
                        nodes.append(node1) # add seed page to nodes
                        adj.append([])
                        count = count + 1
            else:
                query = re.sub(r"\s+", '+', query)
                url = "https://www.google.com/search?q="+query+"&num=35" #Getting 35 results if any duplicates
                raw_page = requests.get(url, headers=header).text
                results = re.findall(r'(?<=<h3 class="r"><a href="/url\?q=).*?(?=&)', str(raw_page))
                RootSet = list(set(results))[0:30] #Provides 30 unique of the 35 we requested above
                for page in RootSet:
```

```
                    if count < seed:
                        node1 = node(count,page,1,1)
                        nodes.append(node1) # add seed page to nodes
                        adj.append([])
                        count = count + 1

            if saveToFile: save_file(rootSetFile, RootSet)
            return RootSet

        '''''
        ################################################
        build first base set
        ################################################
        '''

        def addLinkedPages(rootSet):
            seedIndex = 0
            baseSet1 = []
            opener = URLopener()

            for i in rootSet:
                try:
                    page = opener.open(urllib.parse.unquote(i))
                    soup = BeautifulSoup(page.read(), features='lxml')
                    links = soup.findAll("a", href=True)
                except Exception:
                    continue
                for link in links:
                    if validLink(link["href"]) and getDomain(link["href"]) != getDomain(
    i):
                        if link["href"] in rootSet or link["href"] in baseSet1: # page a
    lready exists in graph
                            for x in nodes:
                                if x.url == link["href"]: linkedNode = x
                            for y in nodes:
                                if y.url == i: linkingNode = y
                            if not linkedNode.pageid in adj[linkingNode.pageid]:
                                adj[linkingNode.pageid].append(linkedNode.pageid)
                        else: # it is a new page
                            baseSet1.append(link["href"])
                            nodeid = len(rootSet) + len(baseSet1) - 1
                            node1 = node(nodeid,link["href"],1,1)
                            nodes.append(node1) # add page to graph
                            adj.append([])
                            for x in nodes:
                                if x.url == link["href"]: linkedNode = x
                            for y in nodes:
                                if y.url == i: linkingNode = y
                            if not linkedNode.pageid in adj[linkingNode.pageid]:
                                adj[linkingNode.pageid].append(linkedNode.pageid)
                            addedPages[seedIndex] = addedPages[seedIndex] + 1
                    if addedPages[seedIndex] >= k: break

                seedIndex = seedIndex + 1
                if seedIndex > seed:
                    break

            if saveToFile: save_file(baseSet1File, baseSet1)
            return baseSet1

        '''''
```

```
229.        ##################################################
230.        Build second base set
231.        ##################################################
232.        '''
233.
234.        def addLinkingPages(rootSet, baseSet1):
235.            seedIndex = 0
236.            baseSet2 = []
237.
238.            for i in rootSet:
239.
240.                query = "link:" + i
241.                limit = k - addedPages[seedIndex]
242.                urls = searchYahoo(query , limit)
243.
244.                for j in urls:
245.                    if validLink(j) and getDomain(j) != getDomain(i):
246.                        if j in rootSet or j in baseSet1 or j in baseSet2: # page alread
    y exists in graph
247.                            for x in nodes:
248.                                if x.url == j: linkingNode = x
249.                            for y in nodes:
250.                                if y.url == i: linkedNode = y
251.                            if not linkedNode.pageid in adj[linkingNode.pageid]:
252.                                adj[linkingNode.pageid].append(linkedNode.pageid)
253.
254.                        else: # it is a new page
255.                            baseSet2.append(j)
256.                            nodeid = len(rootSet) + len(baseSet1) + len(baseSet2) - 1
257.                            node1 = node(nodeid,j,1,1)
258.                            nodes.append(node1) # add page to graph
259.                            adj.append([]) # add link to adjacency matrix
260.                            for x in nodes:
261.                                if x.url == i: linkingNode = x
262.                            for y in nodes:
263.                                if y.url == j: linkedNode = y
264.                            if not linkedNode.pageid in adj[linkingNode.pageid]:
265.                                adj[linkingNode.pageid].append(linkedNode.pageid)
266.                            addedPages[seedIndex] = addedPages[seedIndex] + 1
267.
268.                    if addedPages[seedIndex] >= k: break
269.
270.                seedIndex = seedIndex + 1
271.                if seedIndex > seed:
272.                    break
273.
274.            if saveToFile:
275.                save_file(baseSet2File, baseSet2)
276.                with open(adjfile, 'w') as file: # save adjacency list
277.                    file.writelines('\t'.join(str(j) for j in i) + '\n' for i in adj)
278.
279.            return baseSet2
280.
281.
282.        '''''
283.        ##################################################
284.        Build Neighbourhood Graph
285.        ##################################################
286.        '''
287.
288.        def buildWholeSet():
```

```
289.
290.           rootSet = [] #list of seed pages
291.
292.           if online_search==1:
293.               rootSet = build_root_set(searchQuery)
294.               baseSet1 = addLinkedPages(rootSet)
295.               #baseSet2 = addLinkingPages(rootSet, baseSet1)
296.               wholeSet = rootSet + baseSet1 #+ baseSet2
297.
298.               if saveToFile:
299.                   with io.open(allPages, "w", encoding="utf-
    8") as f: #save neighbourhood graph to file allPages
300.                       for item in wholeSet:
301.                           f.write("%s\n" % item)
302.           else: # load from files
303.               rootFile = open(rootSetFile, "r")
304.               rootSet = rootFile.read().split('\n')
305.
306.               base1File = open(baseSet1File, "r")
307.               baseSet1 = base1File.read().split('\n')
308.
309.               #base2File = open(baseSet2File, "r")
310.               #baseSet2 = base2File.read().split('\n')
311.
312.               allPagesFile = open(allPages, "r")
313.               wholeSet = allPagesFile.read().split('\n')
314.
315.               infile = open(adjfile,'r')
316.               for line in infile:
317.                   adj.append(line.strip().split('\t'))
318.               infile.close()
319.
320.           return (wholeSet, adj)
321.
322.       '''''
323.       ##################################################
324.       Print search results
325.       ##################################################
326.       '''
327.
328.       def print_desccriptions1(nodes,Nbound,k):
329.           count = 0
330.           if(len(nodes)== 0):
331.               print("No pages found. Enter an upperbound (k) higher than", k)
332.           while count < Nbound:
333.               try:
334.                   title, description, image = web_preview(nodes[count].url)
335.                   print(title,'\n', description ,'\n ((authority= {0:1.6f}'.format(nod
    es[count].auth) , " |||   hub= {0:1.6f}".format(nodes[count].hub),'))')
336.                   print('-----------------------------------------------------------
    ------------------------')
337.               #except (ConnectionError, HTTPError, Timeout, TooManyRedirects, InvalidU
    RL, KeyError):
338.               except Exception:
339.                   pass
340.               count = count + 1
341.       '''''
342.       def print_desccriptions(nodes,Nbound,k):
343.
344.           descriptions = []
345.
```

```
            count = 0
            if(len(nodes)== 0):
                print("No pages found. Enter an upperbound (k) higher than", k)
            else:
                print("There are less resulting searched pages than", Nbound)
                while count <= Nbound:
                    print('*****************************************', nodes[count].u
    rl)

                    req = urllib.request.Request(nodes[count].url, headers=header)
                    try:
                        html = urllib2.urlopen(req)
                    except:
                        continue

                    soup = BeautifulSoup(html, "lxml")

                    title = soup.title.text
                    metas = soup.find_all("meta")
                    desc = [ meta.attrs['content'] for meta in metas if 'name' in meta.a
    ttrs and meta.attrs['name'] == 'description' ]

                    if len(title) < 1:
                        title = re.findall(r'^(?:https?:\/\/)?(?:[^@\/\n]+@)?(?:www\.)?(
    [^:\/\n]+)', nodes[count].url)
                        title = title[0]
                    if len(desc) < 1:
                        desc = soup.p.text
                    else:
                        desc = desc[0]

                    if len(desc) > 140:
                        desc = desc[0:140]+'...'

                    descriptions.append([title, nodes[count].url, desc, nodes[count].aut
    h, nodes[count].hub])
                    count = count + 1

                for dis in descriptions:
                    print(dis[0],'\n', dis[1], '\n', dis[2],'\n', "((authority= {0:1.6f}
    ".format(dis[3]) , " |||    hub= {0:1.6f}".format(dis[4],'))'))
                    print('-----------------------------------------------------------
    ------------------------')
        '''



    def printPages(nodes,N,k):

        if printNeighbourhood==1:
            print('\n=============================================')
            print("    pages in the neighbourhood graph")
            print("=============================================")
            print_desccriptions1(nodes,len(nodes),k)

        nodes.sort(key=lambda x: x.auth, reverse=True)
        print('\n=============================================')
        print("    pages sorted by authority")
        print("=============================================")
        print_desccriptions1(nodes,N,k)
```

```
401.
402.            nodes.sort(key=lambda x: x.hub, reverse=True)
403.            print('\n=============================================')
404.            print("    pages sorted by hub")
405.            print("=============================================")
406.            print_desccriptions1(nodes,N,k)
407.
408.        def mapping(x):
409.            return x + 100
410.
411.
412.        def showGraph(nodes, adj):
413.
414.            nodeList= []
415.            edgeList = []
416.            options = {
417.                'node_color': 'lavender',
418.                'node_size': 800,
419.                'width': 1,
420.                'arrowstyle': '-|>',
421.                'arrowsize': 20,
422.            }
423.
424.            G = nx.DiGraph(directed=True)
425.
426.            for node in nodes:
427.                nodeList.append(node.pageid)
428.
429.            for i in range(len(adj)):
430.                for j in adj[i]:
431.                    edgeList.append((i,j))
432.
433.            G.add_nodes_from(nodeList)
434.            G.add_edges_from(edgeList)
435.
436.            nx.draw_networkx(G, arrows=True, with_labels = True, **options)
437.            if saveToFile: plt.savefig("graph.png") # save as png
438.            plt.show()
439.
440.
441.
442.        '''''
443.        ##################################################
444.        hits functions
445.        ##################################################
446.        '''
447.
448.        def converge(nodes, errorrate, auth, auth_prev, hub, hub_prev):
449.            counter = 0;
450.            for i in range(len(nodes)):
451.                if ((abs(auth[i] - auth_prev[i]) < errorrate) and (abs(hub[i] - hub_prev
    [i]) < errorrate)):
452.                    counter = counter + 1;
453.            if counter == len(nodes): #converge if all differences are less than errorra
    te
454.                return 1
455.            else:
456.                return 0
457.
458.        def hits(nodes, adj, errorrate):
459.            converged = 0
```

```
460.            auth = []
461.            auth_prev = []
462.            hub = []
463.            hub_prev = []
464.
465.            for p in nodes:
466.                p.auth = 1
467.                p.hub = 1
468.                auth.append(p.auth)
469.                hub.append(p.hub)
470.                auth_prev.append(p.auth)
471.                hub_prev.append(p.hub)
472.
473.            while True:
474.
475.                auth_prev = auth[:]
476.                hub_prev = hub[:]
477.
478.                #update authority values first
479.                norm = 0
480.                for p in nodes:
481.                    #look in the adj list for incoming links to p
482.                    incoming = []
483.                    for i in range(len(adj)):
484.                        if p.pageid in adj[i]:
485.                            incoming.append(i)
486.
487.                    for q in nodes:    # for q in p.incoming
488.                        if q.pageid in incoming:
489.                            p.auth = p.auth + q.hub
490.                    norm = norm + math.pow(p.auth,2) #calculate the sum of the squared a
    uth values to normalise
491.                norm = math.sqrt(norm)
492.
493.                for p in nodes:
494.                    p.auth = p.auth / norm if norm else 1 #update the auth scores with n
    ormalization
495.
496.                # update hub values
497.                norm = 0
498.                for p in nodes:
499.                    for q in nodes: # for q in p.outgoing
500.                        if q.pageid in adj[p.pageid]:
501.                            p.hub = p.hub + q.auth
502.                    norm = norm + math.pow(p.hub,2) #calculate the sum of the squared hu
    b values to normalise
503.                norm = math.sqrt(norm)
504.                for p in nodes:
505.                    p.hub = p.hub / norm if norm else 1 #update the hub scores with norm
    alization
506.                    #hub_prev[count] = p.hub
507.
508.                for i in range(len(nodes)):
509.                    auth[i] = nodes[i].auth
510.                    hub[i] = nodes[i].hub
511.
512.                converged = converge(nodes, errorrate, auth, auth_prev, hub, hub_prev)
513.                if (converged): break
514.
515.
516.        '''''
```

```
517.        ##################################################
518.        main function
519.        ##################################################
520.        '''
521.
522.        if __name__ == "__main__":
523.
524.            k = int(input("define Upper bound for added pages (k): " or 5 )) # uper boun
    d of added pages for each seed page
525.            N = int(input("define Upper bound for printed pages (N): " or 5 )) # uper bo
    und of added pages for each seed page
526.            wholeSet, adj = buildWholeSet()
527.            hits(nodes, adj, error_rate)
528.            printPages(nodes,N,k)
529.            if printGraph:
530.                showGraph(nodes, adj)
531.
532.
533.
534.
```