IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 12, December 2017

Data Science as an Interdisciplinary Field for Knowledge Discovery

Dr .V. V. Narendra Kumar

Professor CSE/IT, St.Mary's Engineering College, Hyderabad, India

Abstract: Data science has attracted a lot of attention, promising to turn vast amounts of data into useful predictions and insights. Data scientists see data science from three perspectives namely, statistical, computational and human. The effective combination of all three components is the essence of what data science is about. Now a day's Data science is a buzzword used in business, academia and government. Data Science has occupied a prominent role in e-business and has started fetching fruits in online shopping like Amazon, Flipkart etc. In future Data Science will have a prominent role in every walk of human life right from domestic life to official life.

Keywords: Data Science, Big Data, Analytics , e-Business, Online Marketing.

I. INTRODUCTION

Data science[1] can be defined as an interdisciplinary field in which processes and systems are used to extract the hidden knowledge from data. Data scientists collect, manage, analyze and interpret vast amounts of data with a diverse array of applications. For example Amazon recommendations, marketing campaigns, Uber, Siri, price comparison sites, gaming and image recognition are all powered, to varying degrees, by data science. Very recently many industries are realizing the importance of Data Science in their day-to-day decisions. In the wake of mobile advertising, personalization, and an overall need for 'data-savviness', it's clear data science is going to stay forever.

II. DATA SCIENCE PROCESS

Data science[2], also known as data-driven science, is an interdisciplinary field about scientific methods, processes and systems to extract knowledge or insights from data in various forms, structured or unstructured, similar to Knowledge Discovery in Databases (KDD). The data science process used to make decisions is depicted in the Figure 1.

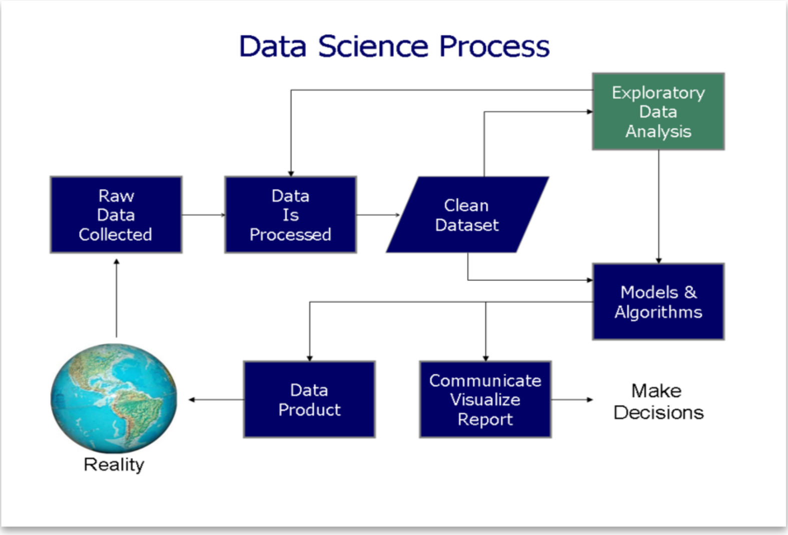


Figure 1. The Data Science Process

Data refers to a collection of facts (numbers, words, measurements, observations, etc) that has been translated into a form that computers can process. Many successful stories in the industry involve "data" in changing the face of our world. In healthcare to cure a disease, in companies boost a company's revenue, in construction to make a building more efficient in marketing to efficiently advertise a product etc. data has a prominent role. In general, data is simply another word for information. But in computing and business (most of what you read about in the news when it comes to data – especially if it's about Big Data), data refers to information that is machine-readable as opposed to human-readable.

DOI 10.17148/IJARCCE.2017.61208

Science refers to a system of acquiring knowledge. This system uses observation and experimentation to describe and explain natural phenomena. The term science also refers to the organized body of knowledge people have gained using that system. Science is the concerted human effort to understand, or to understand better, the history of the natural world and how the natural world works, with observable physical evidence as the basis of that understanding.

"**Results**" Means the Ending of a Scientific Story. If you've ever attended a science fair or heard the explanation of an amazing experiment, you know that science can sometimes seem like a story. A scientific experiment has a beginning and an end.

Hence Data Science can be understood as a system that uses observation and experimentation to describe and explain natural phenomena of acquiring knowledge from data with observable physical evidence as the basis of that understanding with a beginning and an end. This is illustrated in Figure 2.

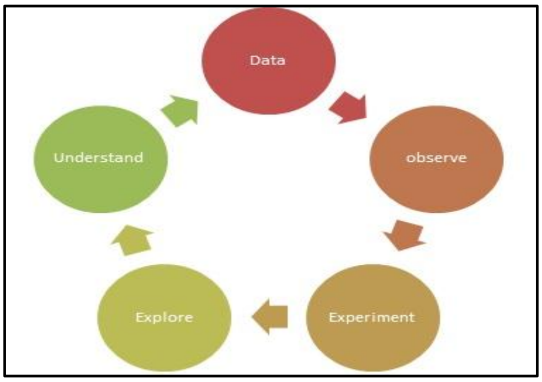


Figure 2. Data Science system

III. POPULAR ALGORITHMS / METHODS AND COMPUTER LANGUAGES USED BY DATA SCIENTISTS

Table 1. Popular Algorithms used in Data Science

SNO	Algorithm
1	Regression
2	Clustering
3	Decision Trees/Rules
4	Visualization
5	K-Nearest Neighbors
6	Principal Component Analysis (PCA)
7	Statistics
8	Random Forests
9	Time Series/Sequence
10	Text Mining

Table 2. Popular Languages of Data Science

SNO	Computer Language
1	R
2	Python
3	Java
4	JavaScript
5	C
6	C++
7	Julia
8	Scala
9	Lua
10	SparK

Industry Data Scientists are mostly use Regression, Visualization, Statistics, Random Forests, and Time Series. Government/non-profit are more likely to use Visualization, PCA, and Time Series. Academic researchers are more likely to use PCA and Deep Learning. Students generally use fewer algorithms, but do more text mining and Deep Learning (Table 1). Nearly 10 popular Computer languages can be widely used to analyze these algorithms (Table 2). Here I would like to mention an algorithm which is used in IoT , RFID applications of electronics industry.

Scapegoat Tree

A Scapegoat [3] tree is a self-balancing Binary Search Tree like AVL Tree, Red-Black Tree, Splay Tree,etc.

- Search time is O(Log n) in worst case. Time taken by deletion and insertion is amortized O(Log n)
- The balancing idea is to make sure that nodes are α size balanced. A size balanced means sizes of left and right subtrees are at most α * (Size of node). The idea is based on the fact that if a node is A weight balanced, then it is also height balanced: height <= log1/&alpha;(size) + 1

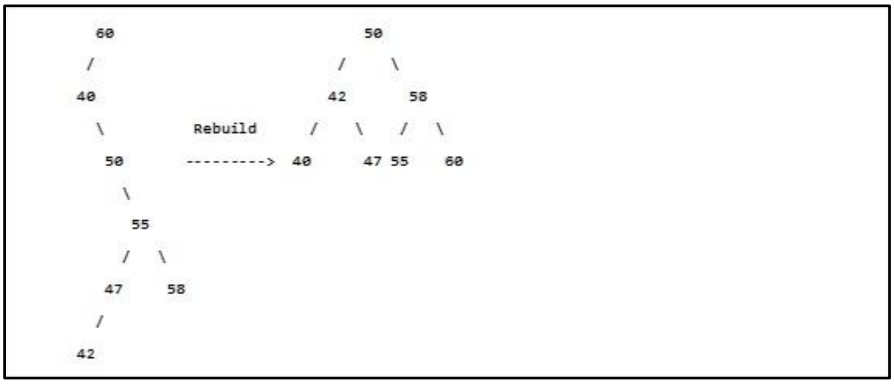
DOI 10.17148/IJARCCE.2017.61208

- Unlike other self-balancing BSTs, Scapegoat tree doesn’t require extra space per node. For example, Red Black Tree nodes are required to have color. In below implementation of Scapegoat Tree, we only have left, right and parent pointers in Node class. Use of parent is done for simplicity of implementation and can be avoided.

Insertion (Assuming α = 2/3):
To insert value x in a Scapegoat Tree:

- Create a new node u and insert x using the BST insert algorithm.
- If the depth of u is greater than log3/2n where n is number of nodes in tree then we need to make tree balanced. To make balanced, we use below step to find a scapegoat.
- Walk up from u until we reach a node w with size(w) > (2/3)*size(w.parent). This node is scapegoat
- Rebuild the subtree rooted at w.parent.

What does rebuilding the subtree mean?
In rebuilding, we simply convert the subtree to the most possible balanced BST. We first store inorder traversal of BST in an array, then we build a new BST from array by recursively dividing it into two halves.



```
  60                            50
 /                           /     \
40                          42       58
  \          Rebuild      /    \    /   \
   50       --------->  40      47 55    60
     \
      55
     /   \
   47     58
  /
42
```

Java Program to Implement Scapegoat Tree
This is a Java Program to implement Scapegoat Tree. A scapegoat tree is a self-balancing binary search tree which provides worst-case O(log n) lookup time, and O(log n) amortized insertion and deletion time.
Here is the source code of the Java program to implement Scapegoat tree. The Java program is successfully compiled and run on a Windows system. The program output is also shown below.

```
/*
 * Java Program to Implement Scapegoat Tree
 */

import java.util.Scanner;

/* Class SGTNode */
class SGTNode
{
    SGTNode right, left, parent;
    int value;

    /* Constructor */
    public SGTNode(int val)
    {
        value = val;
    }
}

/* Class ScapeGoatTree */
class ScapeGoatTree
{
```

DOI 10.17148/IJARCCE.2017.61208

```
private SGTNode root;
private int n, q;

/* Constructor */
public ScapeGoatTree()
{
    root = null;
    // size = 0
    n = 0;
}
/* Function to check if tree is empty */
public boolean isEmpty()
{
    return root == null;
}
/* Function to clear  tree */
public void makeEmpty()
{
    root = null;
    n = 0;
}
/* Function to count number of nodes recursively */
private int size(SGTNode r)
{
    if (r == null)
        return 0;
    else
    {
        int l = 1;
        l += size(r.left);
        l += size(r.right);
        return l;
    }
}
/* Functions to search for an element */
public boolean search(int val)
{
    return search(root, val);
}
/* Function to search for an element recursively */
private boolean search(SGTNode r, int val)
{
    boolean found = false;
    while ((r != null) && !found)
    {
        int rval = r.value;
        if (val < rval)
            r = r.left;
        else if (val > rval)
            r = r.right;
        else
        {
            found = true;
            break;
        }
        found = search(r, val);
    }
    return found;
}
```

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 12, December 2017

```
/* Function to return current size of tree */
public int size()
{
   return n;
}
/* Function for inorder traversal */
public void inorder()
{
   inorder(root);
}
private void inorder(SGTNode r)
{
   if (r != null)
   {
      inorder(r.left);
      System.out.print(r.value +" ");
      inorder(r.right);
   }
}

/* Function for preorder traversal */
public void preorder()
{
   preorder(root);
}
private void preorder(SGTNode r)
{
   if (r != null)
   {
      System.out.print(r.value +" ");
      preorder(r.left);
      preorder(r.right);
   }
}

/* Function for postorder traversal */
public void postorder()
{
   postorder(root);
}
private void postorder(SGTNode r)
{
   if (r != null)
   {
      postorder(r.left);
      postorder(r.right);
      System.out.print(r.value +" ");
   }
}
private static final int log32(int q)
{
   final double log23 = 2.4663034623764317;
   return (int)Math.ceil(log23*Math.log(q));
}
/* Function to insert an element */
public boolean add(int x)
{
   /* first do basic insertion keeping track of depth */
   SGTNode u = new SGTNode(x);
```

DOI 10.17148/IJARCCE.2017.61208

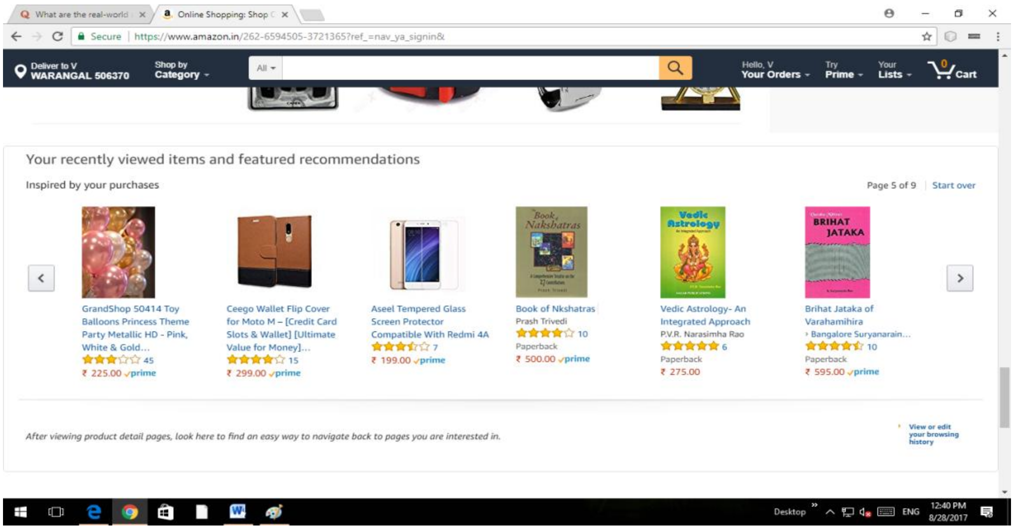
```
    int d = addWithDepth(u);
    if (d > log32(q)) {
        /* depth exceeded, find scapegoat */
        SGTNode w = u.parent;
        while (3*size(w) <= 2*size(w.parent))
            w = w.parent;
        rebuild(w.parent);
    }
    return d >= 0;
}
/* Function to rebuild tree from node u */
protected void rebuild(SGTNode u)
{
    int ns = size(u);
    SGTNode p = u.parent;
    SGTNode[] a = new SGTNode[ns];
    packIntoArray(u, a, 0);
    if (p == null)
    {
        root = buildBalanced(a, 0, ns);
        root.parent = null;
    }
    else if (p.right == u)
    {
        p.right = buildBalanced(a, 0, ns);
        p.right.parent = p;
    }
    else
    {
        p.left = buildBalanced(a, 0, ns);
        p.left.parent = p;
    }
}
/* Function to packIntoArray */
protected int packIntoArray(SGTNode u, SGTNode[] a, int i)
{
    if (u == null)
    {
        return i;
    }
    i = packIntoArray(u.left, a, i);
    a[i++] = u;
    return packIntoArray(u.right, a, i);
}
/* Function to build balanced nodes */
protected SGTNode buildBalanced(SGTNode[] a, int i, int ns)
{
    if (ns == 0)
        return null;
    int m = ns / 2;
    a[i + m].left = buildBalanced(a, i, m);
    if (a[i + m].left != null)
        a[i + m].left.parent = a[i + m];
    a[i + m].right = buildBalanced(a, i + m + 1, ns - m - 1);
    if (a[i + m].right != null)
        a[i + m].right.parent = a[i + m];
    return a[i + m];
}
/* Function add with depth */
```

```
    public int addWithDepth(SGTNode u)
    {
        SGTNode w = root;
        if (w == null)
        {
            root = u;
            n++;
            q++;
            return 0;
        }
        boolean done = false;
        int d = 0;
        do {

            if (u.value < w.value)
            {
                if (w.left == null)
                {
                    w.left = u;
                    u.parent = w;
                    done = true;
                }
                else
                {
                    w = w.left;
                }
            }
            else if (u.value > w.value)
            {
                if (w.right == null)
                {
                    w.right = u;
                    u.parent = w;
                    done = true;
                }
                w = w.right;
            }
            else
            {
                return -1;
            }
            d++;
        } while (!done);
        n++;
        q++;
        return d;
    }
}

public class ScapeGoatTreeTest
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        /* Creating object of ScapeGoatTree */
        ScapeGoatTree sgt = new ScapeGoatTree();
        System.out.println("ScapeGoat Tree Test\n");
        char ch;
        /*  Perform tree operations  */
        do
        {
            System.out.println("\nScapeGoat Tree Operations\n");
            System.out.println("1. insert ");
            System.out.println("2. count nodes");
            System.out.println("3. search");
            System.out.println("4. check empty");
            System.out.println("5. make empty");

            int choice = scan.nextInt();
            switch (choice)
            {
            case 1 :
                System.out.println("Enter integer element to insert");
                sgt.add( scan.nextInt() );
                break;
            case 2 :
                System.out.println("Nodes = "+ sgt.size());
                break;
            case 3 :
                System.out.println("Enter integer element to search");
                System.out.println("Search result : "+ sgt.search( scan.nextInt() ));
                break;
            case 4 :
                System.out.println("Empty status = "+ sgt.isEmpty());
                break;
            case 5 :
                System.out.println("\nTree cleared\n");
                sgt.makeEmpty();
                break;
            default :
                System.out.println("Wrong Entry \n ");
                break;
            }
            /*  Display tree  */
            System.out.print("\nPost order : ");
            sgt.postorder();
            System.out.print("\nPre order : ");
            sgt.preorder();
            System.out.print("\nIn order : ");
            sgt.inorder();
            System.out.println("\nDo you want to continue (Type y or n) \n");
            ch = scan.next().charAt(0);
        } while (ch == 'Y'|| ch == 'y');
    }
}
```

IV. APPLICATIONS / USES OF DATA SCIENCE

Data Science algorithms[4] are widely used in

- Search engines like Google, Yahoo, Bing etc. to deliver the best result for our searched query in fraction of seconds
- digital marketing from displaying of banners on various websites to the digital bill boards at the airports
- Amazon, Twitter, Google Play, Linkedin etc. use to promote their products / suggestions in accordance with user’s interest and relevance of information (Figure 4).



Your recently viewed items and featured recommendations

Inspired by your purchases

GrandShop 50414 Toy Balloons Princess Theme Party Metallic HD - Pink, White & Gold... 45 ₹ 225.00 prime

Ceego Wallet Flip Cover for Moto M – [Credit Card Slots & Wallet] [Ultimate Value for Money]... 15 ₹ 299.00 prime

Aseel Tempered Glass Screen Protector Compatible With Redmi 4A 7 ₹ 199.00 prime

Book of Nkshatras Prash Trivedi 10 Paperback ₹ 500.00 prime

Vedic Astrology- An Integrated Approach P.V.R. Narasimha Rao 6 Paperback ₹ 275.00

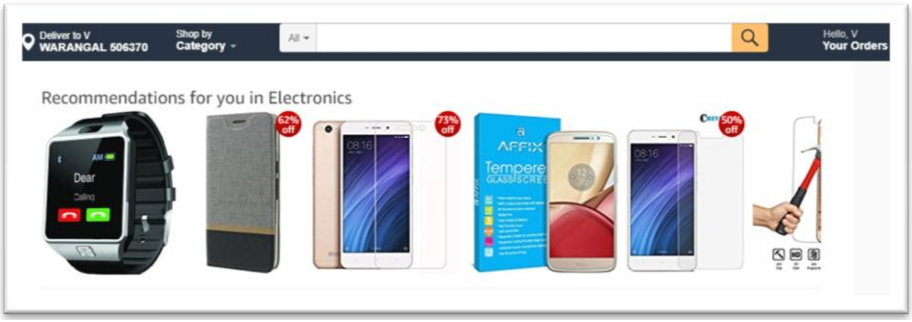
Brihat Jataka of Varahamihira › Bangalore Suryanarain... 10 Paperback ₹ 595.00 prime

After viewing product detail pages, look here to find an easy way to navigate back to pages you are interested in.

Figure 4. The authors Amazon suggestions page

- Facebook for getting suggestions and to tag your friends using face recognition algorithm.
- WhatsApp web to scan a barcode in your web browser using mobile phone using image recognition
- Speech recognition products are Google Voice, Siri, Cortana etc.
- Computer games such as EA Sports, Zynga, Sony, Nintendo, Activision-Blizzard to leed to gaming experience to the next level using machine learning algorithms
- price comparison websites PriceGrabber, PriceRunner, Junglee, Shopzilla, DealTime are being driven by lots and lots of data which is fetched using APIs and RSS Feeds
- Airline Industry to Predict flight delay, Decide which class of airplanes to buy, Whether to directly land at the destination, or take a halt in between
- One of the first applications of data science originated from Finance discipline. Companies were fed up of bad debts and losses every year. However, they had a lot of data which use to get collected during the initial paper work while sanctioning loans. They decided to bring in data science practices in order to rescue them out of losses. Over the years, banking companies learned to divide and conquer data via customer profiling, past expenditures and other essential variables to analyze the probabilities of risk and default. Moreover, it also helped them to push their banking products based on customer's purchasing power.
- Logistic companies like DHL, FedEx, UPS, use data science to improve their operational efficiency. Using data science, these companies have discovered the best routes to ship, the best suited time to deliver, the best mode of transport to choose thus leading to cost efficiency, and many more to mention.
- In projects like self-driving cars project, Robots etc
- Many applications in marketing, logistics, fraud detection and customer service. To develop predictive models for e-commerce marketing to

○ Predictive churn rate: To identify active, at-risk and lost customers which helps customize marketing
○ Predictive customer lifetime value: Gives an estimation on how much you can expect to earn from a customer over his lifetime. This is useful in knowing how much you should spend on acquisition
○ Replenishment: Identify the right time that a customer will need to reorder a product again
○ Recommendations: Suggest products that customer are most likely to buy, based on his purchase history or on the product he is currently viewing (Figure 5)
○ Affinity analysis: Used to identify groups of products that are bought together



Recommendations for you in Electronics

Figure 5. The Author's recommendations page of Amazon

DOI 10.17148/IJARCCE.2017.61208

Apart from the applications mentioned above, data science is also used in Marketing, Finance, Human Resources, Health Care, Government Policies and every possible industry where data gets generated. Using data science, the marketing departments of companies decide which products are best for Up selling and cross selling, based on the behavioral data from customers. In addition, predicting the wallet share of a customer, which customer is likely to churn, which customer should be pitched for high value product and many other questions can be easily answered by data science. Finance (Credit Risk, Fraud), Human Resources (which employees are most likely to leave, employees performance, decide employees bonus) and many other tasks are easily accomplished using data science in these disciplines.

V. THE FUTURE OF DATA SCIENCE

The future of Data Science[5] is very bright. Data Science has immense applications in industries including finance, energy, travel, and government. Many universities started to recognize the importance of Data Science and are offering courses and programs in this field. Campus wide initiatives have increased the opportunity for students to study this field and develop a career in data science. These courses helps learners gain a better understanding of basic concepts in data science and learn the fundamentals of statistics, machine learning and algorithms. Job trends show that the need for data scientists is growing at a fast rate.The industry is facing a challenge of filling these positions with qualified and skilled professionals. According to a recent study conducted by Wanted Analytics, only 4% of the 332,000 computer programmers in the United States currently have the skill set required for data science. And global management consulting firm McKinsey & Company forecasts by 2018 there will be 4 million big data-related jobs in the US, and a shortage of 140,000 to 190,000 data scientists.

REFERENCES

[1.] International Journal on Data Science and Technology
[2.] Data Science Journal
[3.] A. Al-Rawi , A. Lansari , F. Bouslama, A new non-recursive algorithm for binary search tree traversal, Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003.
[4.] International Journal of Current Trends in Science and Technology
[5.] International Journal of Data Science and Analytics

DOI 10.17148/IJARCCE.2017.61208