Review of Dynamic File Classification Solution Using Decision Tree Algorithm

Francis Bambo[1], Deepak Pathak[2], Gagan Sharma[2]

[1] *M.Tech Student, Sri Satya Sai College of Engineering, RKDF University, Bhopal*

[2,3]*Assistant Professor, Mechanical Engineering, Sri Satya Sai College of Engineering, RKDF University, Bhopal*

Corresponding Author: Francis Bambo Manuscript Received: Manuscript Accepted:

Abstract

File Classification approach is an important part for all operating systems as well as Business and end users. In this paper, we semi-automate the process of manual file classification problems based on the file attributes (name, type, date and size), either one level or multi-level automation .DFCS applies a decision tree searching, IF THEN rule based , LINQ and mining techniques to achieve the challenging searching and file classification tasks and also applied threading and background worker techniques which help in speeding up the process of it. DFCS applies MD5 hashing algorithm to find the duplicated files as a preprocess of mining requirements. In experiments, DFCS has much higher precision than SFCS and is comparable with other file classification system like ABBYY Smart Classifier and RoboBasket application. The experiments show encouraging results for the tested files in our systems.

I. INTRODUCTION

An implant is a medical device manufactured to replace a missing biological structure, to support a damaged biological structure or to improve an existing biological structure. Medical implants are artificial devices, as opposed to a transplant, which is transplanted biomedical tissue. The surface of the implants that come into contact with the body can be made with a biomedical material such as titanium, silicone or apatite, whichever is more functional [1]. In some cases, the systems contain electronic components, e.g. artificial pacemakers and cochlear implants. Some implants are bioactive, such as subcutaneous drug delivery devices in the form of implantable pills or drug eluting stents. Some of the examples are given in figure 1.1.

I. INTRODUCTION TO FILE CLASSIFICATION

Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret [1].

File classification is the process of putting files into categories based on their features and properties. Files can be classified either manually or automatically.

Manual classification of large numbers of Files has the following problems associated with it:

- It is labor-intensive (needs lots of work).

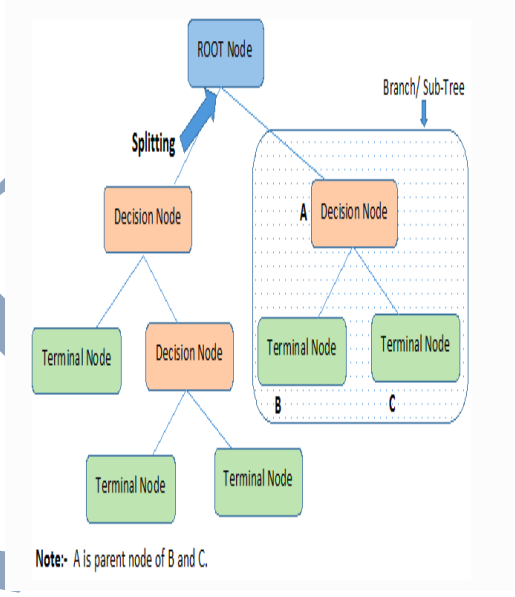
- It is expensive because it requires a lot of classification specialists.
- It is slow and cannot be used in projects where time is of the essence.
- Classification quality deteriorates when classification specialists have to work to tight deadlines.

In this paper (DFCS), you can classify Files automatically, avoiding most of the problems associated with manual file classification. File classification uses the standard technology to classify the files. It can be easily integrated into files management systems, knowledge bases, and other prediction systems that work with files [1] (Hastie TJ).

II. DECISION TREE ALGORITHM

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. [3] (Patel N).

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data).
In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.



III. TECHNICAL ASPECT OF FILE CLASSIFICATION

Benefits of Dynamic File Classification using Decision Tree:

File classification system can be used to classify files in many different types, assigning these files to their appropriate categories as instructed by the user. When classifying files, file classification looks for certain features which are characteristic of each category of files? In the case of (Type, Sub Type, Size, Creation Date, Last Modified, etc.). Standard classification technology and user needs can be applied in order to classify files based on (Type, Sub Type, Size, Creation Date, Last Modified, etc..). [4] (Berry MJA)

DFCS offers an excellent graphical user interface which is similar to the more standard desktop application now a days such (Microsoft Office 2007 and above interfaces), which is familiar to any computer end users and does not require any special skills from the end user.

File classification system can be used in a wide range of tasks that require to search among vast amounts of unsorted (classified) files , such as classify network shared folder , download folders , and other scattered and repeated files in our hard disk to their appropriate categories which help in organizing our computer as well as reducing the time of doing that kind of classification manually, finding and removing duplicate files which helps in making free spaces in our computer hard disk as well as reducing the time and iteration through files in order to classify them, helps in prediction the new file classification category by comparing that files against the already classified files and many more.

File classification system can automate a lot of classification chores (daily work), simplifying a lot of business processes and enabling employees to navigate their way through enormous masses of files. By automating their files classification routines, companies will radically speed up file processing and avoid human error (duplication) that are almost inevitable (occur) when large volumes of files are classified manually.

File classification system can give you an easy way to draw the charts of the classifying which gives you a general idea of the files which are stores in your computer.[13] (SAS Enterprise miner 12.1).

IV. REQUIREMENT ANALYSIS

Hardware Requirements

Minimum requirement

- Hard disk: 40 GB.
- RAM: 512 MB.
- Processor Speed: 3.00GHz.
- Processor: Pentium IV Processor

Software Requirements specification

The software requirement specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description as functional representation, a representation of system behavior, an indication of performance requirements and design constraints, appropriate validation criteria.

A software requirements specification is developed as a consequence of analysis. Review is essential to ensure that the developer and customers have the same perception.

Software requirements [18](Loh WY) specification (SRS) is the starting point of the software development activity. The Software Requirements Specification is produced at the culmination of the analysis task. The introduction of the software requirements specification states the goals and objectives of the software, describing it in the context of the computer-based system. The software requirements specification includes an information description, functional description, behavioral description, validation criteria.

The purpose of this document is to present the software requirements in a precise and easily understood manner. This document provides the functional, performance, design and verification requirements of the software to be developed.

As systems engineering, a requirement can be a description of what a system must do, referred to as a Functional Requirement. This type of requirement specifies something that the delivered system must be able to do. Another type of requirement specifies something about the system itself, and how well it

performs its functions. Such requirements are often called Nonfunctional requirements, or 'performance requirements' or 'quality of service requirements.' Examples of such requirements include usability, availability, reliability, supportability, testability and maintainability.[6] (Zebran MF)

A collection of requirements defines the characteristics or features of the desired system. A 'good' list of requirements as far as possible avoids saying how the system should implement the requirements, leaving such decisions to the system designer. Specifying how the system should be implemented is called "implementation bias" or "solution engineering". However, implementation constraints on the solution may validly be expressed by the future owner.

- MS Operating System or Linux
- Python 3 installed in the system
- C# & VB Languages
- Database Management system
- UI IDL platforms

Functional Requirements:

Functional requirements define a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. [11] (Bhukya DP)

The Functional Requirements which are identified in this project are as follows:

- Login.
- Start level file classification.
- Start multi-level file classification.
- Start file prediction.
- Find duplicate files.
- Exploring files.
- Reporting.

Non-Functional Requirements

Non-Functional requirements describe how the product should be implemented. A Non-Functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. [7][9] (Brimen L, Kass GV) Non-Functional requirements are often called qualities of a system [11] (Bhukya DP). The major Non-Functional requirements of the system area as follows:

Usability

The system is designed with completely automated process hence there is no or less user intervention. The end user can easily navigate the entire system as it is developed in web application. The application gives the status messages regularly based on the user actions performed. Thus, the access to this system is very easy.

Reliability

The system is more reliable because of the qualities that are inherited C# language. This application does not depend on the external memory and hence its space complexity is very less.

V. CONCLUSION AND FUTURE WORK

Conclusion

In this research an approach was proposed to overcome the statically traditional file classification using the power and ability of machine learning classification algorithms to solve such problem.

This research solution is implemented in desktop application development. Visual basic.net language is used as front end. The user interface is designed using Dotnetbar dll library, in such a way like that interfaces being seen in Microsoft office which is very flexible and can be easily navigated and accessed by the end users. python is used for the ML work; MS Access is used as back end for storing the system data.

Performance

This solution App is developing in the high-level languages and using the advanced front-end technology it will give response to the end user on client system with in very less time with accuracy. This application also reduces the energy consumption and also save the utilization of bandwidth.

According to our algorithms, we used MD5 encoding algorithms to find out the duplicated files in the system Thus, the MD5 helps in reducing the time of comparing as well as the space of the system. We design a new algorithm for classifying the files according to their features such as types, date, name, and size in to their proper folders based on supervised learning.

We used threading, background workers techniques , thus help in speeding up our algorithms and make our algorithms showing high performance result than other application such as ABBYY ,and Robo basket , but ABBYY got high accuracy in the file classification by name and content which our algorithms still not producing high accuracy in it.

Finally, our algorithm showing a high accuracy than other application in predicting the file class, which means putting the new file(s) in its appropriate classification folder.

Future work:

In our research, we find that the Dynamic file classification is future of the dealing and working with the huge interconnected and complication files in the development or industrial environment. More methods, technologies and ML &DL algorithms are need in the future work to improve current approach.

REFERENCES

[1] Hastie TJ, Tibshirani RJ, Friedman JH. The Elements of Statistical Learning: Data Mining Inference and Prediction. Second Edition. Springer; 2009. ISBN 978-0-387-84857-0 [Google Scholar]
[2] Fallon B, Ma J, Allan K, Pillhofer M, Trocmé N, Jud A. Opportunities for prevention and intervention with young children: lessons from the Canadian incidence study of reported child abuse and neglect. Child Adolesc Psychiatry Ment Health. 2013;7: 4. [PMC free article] [PubMed] [Google Scholar]
[3] Patel N, Upadhyay S. Study of various decision tree pruning methods with their empirical comparison in WEKA. Int J Comp Appl. 60(12):20–25. [Google Scholar]
[4] Berry MJA, Linoff G. Mastering Data Mining: The Art and Science of Customer Relationship Management. New York: John Wiley & Sons, Inc; 1999. [Google Scholar]
[5] Hastie T, Tibshirani R, Friedman J.The Elements of Statistical Learning. Springer; 2001. p. 269-272 [Google Scholar]
[6] Zibran MF. Department of Computer Science. Diagnostic and Statistical Manual of Mental Disorders – Fourth Edition. Alberta, Canada: Department of Computer Science, University of Calgary; 2012. [Google Scholar]
[7] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees. Belmont

California: Wadsworth, Inc.; 1984. [Google Scholar]

[8] Quinlan RJ. C4.5: Programs for Machine Learning. San Mateo California: Morgan Kaufmann Publishers, Inc.; 1993. [Google Scholar]

[9] Kass GV. Anexploratory technique for investigating large quantities of categorical data. Appl Stat. 1980;29: 119–127. [Google Scholar]

[10] Loh W, Shih Y. Split selection methods for classification trees. Statistica Sinica. 1997;7: 815–840. [Google Scholar]

[11] Bhukya DP, Ramachandram S. Decision tree induction-an approach for data classification using AVL–Tree. Int J Comp d Electrical Engineering. 2010;2(4): 660–665. doi: [10].7763/IJCEE. 2010.V2.208. [CrossRef] [Google Scholar]

[12] Lin N, Noe D, He X. Tree-based methods and their applications in: Pham H. Springer Handbook of Engineering Statistics. London: Springer-Verlag; 2006. p. 551-570. [Google Scholar]

[13] SAS Institute Inc. SAS Enterprise Miner12.1 Reference Help, Second Edition. USA: SAS Institute Inc; 2011. [Google Scholar]

[14] IBM Corporation. IBM SPSS Modeler 17 Modeling Nodes. USA: IBM Corporation; 2015. [Google Scholar]

[15] Is See5/C5.0 Better Than C4.5? Australia: Rulequest Research; 2011. [[updated 2008 Oct; cited 2015 April]]. Available from: http://rulequest.com/see5-comparison.html . [Google Scholar]

[16] IBM Corporation. IBM SPSS Statistics 23 Command Syntax Reference. USA: IBM Corporation; 2015. [Google Scholar]

[17] Batterham PJ, Christensen H, Mackinnon AJ. Modifiable risk factors predicting major depressive disorder at four-year follow-up: a decision tree approach. BMC Psychiatry. 2009;9:75. doi: 10.1186/1471-244X-9-75. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

[18] Loh WY. Fifty years of classification and regression trees. Int Stat Rev. 2014;82(3): 329–348. doi: 10.1111/insr.12016. [PMC free article] [PubMed] [CrossRef] [Google Scholar]