Web Analytics

Understanding user behavior and improving web site performance

Based on analytics derived using AI

Thesis Defense by Jagadeesh Ambati

Guide: Dr. Paul Benjamin

Acknowledgement

I wish to thank *Dr. Paul Benjamin* for providing me the opportunity to work under him. His vision, knowledge, patience are the key factors in completing my thesis. I have to thank him for replying my countless emails with questions, doubts, requests.

I thank my *parents* for their support and strength they have given me during all times. And I have to thank my friends (*Matt, Cathy Zura, Badal Mehrotra, Vikas Srivastava, Kamesh Konchada, Nagesh Nadella and many others*) who have put up with my questions, emails, requests for resources (like keeping the apartment lights on all night). Their patience, jokes have been a key part in completing my thesis.

I sincerely thank *CSIS Faculty, Staff and Students* for all their direct and indirect help.

Thank You all for your support.

Sincerely,

Jagadeesh Ambati

CONTENTS

ABSTRACT

The information on the internet is acquired through surfing the web, searching on the web using search engines. The keywords used by people are the vital link to the information they need. The retrieval of the web pages by the search engines largely depends on various search technologies, largely proprietary technologies. The returned page if it matches the users needs or not depends on the user himself. As a result the search engines based on their generic search algorithms retrieve the web pages based on the keywords used by the user, the occurrence of the key word on the web page, meta tags, links pointing to that page etc. Many at times we often click on the page returned by the search engine to find what we are looking for. But because of web pages becoming too lengthy, complex, big, we often end up searching on the returned page itself.

Web sites have been developed using latest technologies, large amount of time, money and effort has been invested by entities using them. As a result the knowledge about whether the site is user friendly or search engine friendly is not being used to make the site more efficient and decrease the time spent by the administrator in making it user friendly.

Corporate sites, online store web sites, and educational sites are information banks which need to give value to the web site users at any given time. When an internet surfer searches for information on the web and reaches a website, it means that he/she has spent considerable time and effort in trying to find what is needed. Many web sites invest in personalization technologies where the user takes the time and effort to personalize what he or she needs. But the personalization done by the web sites for the user/users at their end is less. At any level, it's the user who himself has to tell the

web site what he or she needs. But trying to know what a web surfer wants has been an issue and many people have succeeded. The pop-ups, emails all are because our trail has been picked up and the smartest entity wants to offer us what we want (in most cases it’s not what the surfer wants). As a result we use anti popup software's, spam guards in our email so that we aren't bothered.

But what is the best way to understand the web site surfers, provide them with some time saving tools when they come to the website through a search engine? Is there a way where we can 'learn' about the users and provide them the best tools and also make the life of website administrator easier?

In this thesis, the server log files are used to understand the web surfers by using Artificial Intelligence technologies. Machine Learning (Decision trees) is being used to gain an insight into web surfers by understanding the keywords and the users' behavior on the web site. This an approach which is being used to create a personalization for website users at the website end who reach the web site through keywords, make the web site more user friendly, easy to administer and search friendly. The end web application has been designed to save time for the website surfer and also making it easier for the site administrator to understand new users on the site, their keywords and improve the website efficiency without worrying about taking a look at the lengthy server log files. The web analytics derived from the server log files give us an in-depth knowledge about the web surfers and their activities.

Description

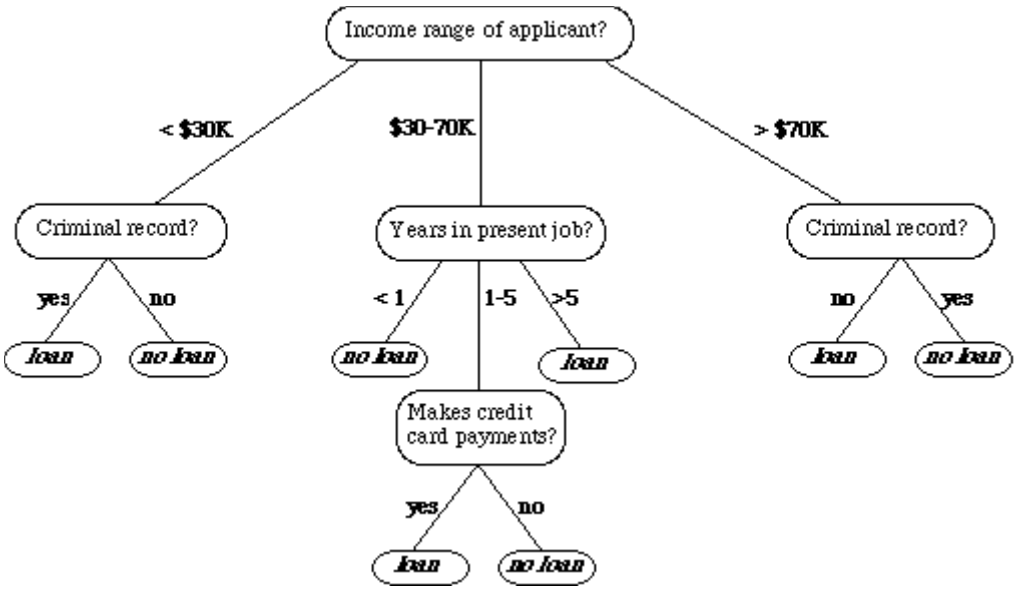
Humankind has given itself the scientific name homo sapiens- man the wise - because our mental capacities are so important to our everyday lives and our sense of self. The field of artificial intelligence or AI attempts to understand intelligent entities. But unlike philosophy and psychology, which are also concerned with intelligence, AI strives to build intelligent entities as well as understand them.

Machine Learning, Data Mining and Data Warehousing

Machine learning is flexible methods for capturing the relationships in data - "summarization" or "compression" into a relational form (**tree/rule** styles, etc) or function approximation (neural nets, etc)

For example we investigate learning from structured databases (for applications such as screening loan applicants), image data (applications such as character recognition), and text collections (for applications such as locating relevant sites on the World Wide Web). For example, we might have

a **decision tree** to help a financial institution decide whether a person should be offered a loan:



Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified.

Data are any facts, numbers, or text that can be processed by a computer. Today, organizations are accumulating vast and growing amounts of data in different formats and different databases. This includes:

- operational or transactional data such as, sales, cost, inventory, payroll, and accounting
- no operational data, such as industry sales, forecast data, and macro economic data
- meta data - data about the data itself, such as logical database design or data dictionary definitions

The patterns, associations, or relationships among all this *data* can provide *information*. For example, analysis of retail point of sale transaction data can yield information on which products are selling and when.

Information can be converted into *knowledge* about historical patterns and future trends. For example, summary information on retail supermarket sales can be analyzed in light of promotional efforts to provide knowledge of consumer buying behavior. Thus, a manufacturer or retailer could determine which items are most susceptible to promotional efforts.

Dramatic advances in data capture, processing power, data transmission, and storage capabilities are enabling organizations to integrate their various databases into *data warehouses*. Data warehousing is defined as a process of centralized data management and retrieval. Data warehousing, like data mining, is a relatively new term although the concept itself has been around for years. Data warehousing represents an ideal vision of maintaining a central repository of all organizational data. Centralization of data is needed to maximize user access and analysis. Dramatic technological advances are making this vision a reality for many companies. And, equally dramatic advances in data analysis software are allowing users to access this data freely. The data analysis software is what supports data mining.

Uses of Data Mining

Data mining is primarily used today by companies with a strong consumer focus - retail, financial, communication, and marketing organizations. It enables these companies to determine relationships among "internal" factors such as price, product positioning, or staff skills, and "external" factors such as economic indicators, competition, and customer demographics. And, it enables them to determine the impact on sales, customer satisfaction, and corporate profits. Finally, it enables them to "drill down" into summary information to view detail transactional data.

With data mining, a retailer could use point-of-sale records of customer purchases to send targeted promotions based on an individual's purchase history. By mining demographic data from comment or warranty cards, the retailer could develop products and promotions to appeal to specific customer segments.

For example, Blockbuster Entertainment mines its video rental history database to recommend rentals to individual customers. American Express can suggest products to its cardholders based on analysis of their monthly expenditures.

Learning Decision Trees

Decision tree induction is one of the simplest and yet most successful forms of learning algorithm.

A decision tree takes an input an object or situation described by a set of properties, and outputs a yes/no "decision". Each internal node in the tree corresponds to a test of the value of one of the properties, and the branches from the node are labeled with the possible values of the test. Each leaf node in the tree specifies the Boolean value to be returned if that leaf is returned.

When to consider using decision trees

Decision trees are to be used when the instances are described by Attribute-Value Pairs. For example when instances are described by a fixed set of attributes like (temperature) and values (hot). Decision trees can be used when the training data is possibly noisy (in correct data: label errors or attribute errors) and when the function is discrete valued.

For example: Decision trees are widely used in the equipment or medical diagnosis. Today decision trees are widely used in risk analysis for credit, loans, insurance, consumer fraud,

employee fraud and modeling calendar scheduling preferences (predicting quality of candidate time)

Decision Tree Representation

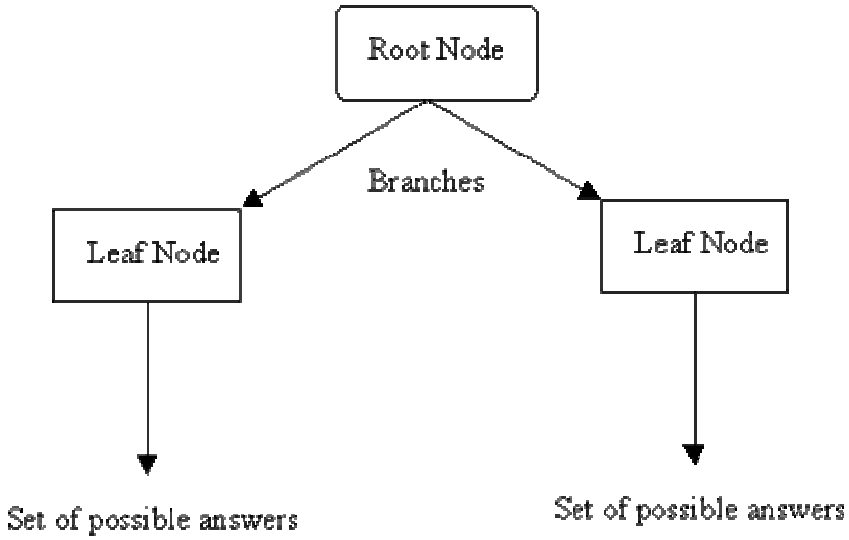
A **decision tree** is an arrangement of tests that prescribes an appropriate test at every step in an analysis. In general, decision trees represent a disjunction of conjunctions of constraints on the attribute-values of instances. Each path from the tree root to a leaf corresponds to a conjunction of attribute tests and the tree itself to a disjunction of these conjunctions.

More specifically, decision trees classify **instances** by sorting them down the tree from the **root node** to some **leaf node**, which provides the classification of the instance. Each node in the tree specifies a **test** of some **attribute** of the instance, and each **branch** descending from that node corresponds to one of the possible **values** for this attribute.

An instance is classified by starting at the root node of the decision tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute. This process is then repeated at the node on this branch and so on until a leaf node is reached.

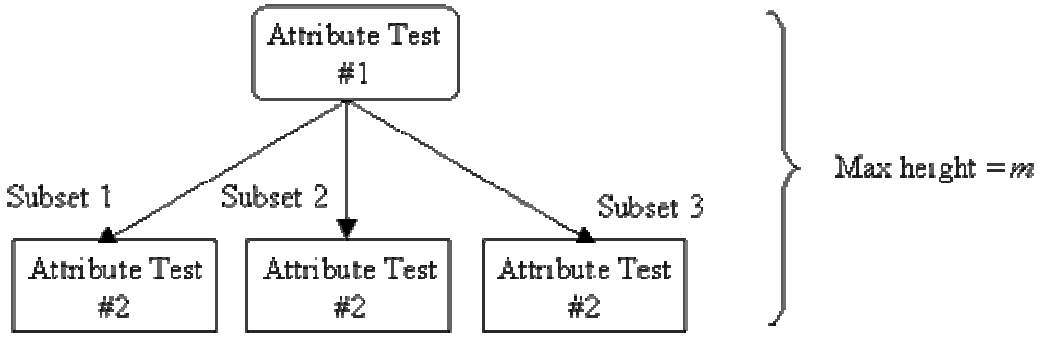
Diagram

- Each non leaf node is connected to a test that splits its set of possible answers into subsets corresponding to different test results.
- Each branch carries a particular test result's subset to another node.
- Each node is connected to a set of possible answers.



Occam's Razor (Specialized to Decision Trees)

"The world is inherently simple. Therefore the smallest decision tree that is consistent with the samples is the one that is most likely to identify unknown objects correctly."



Given *m* attributes, a decision tree may have a maximum height of *m*.

A decision tree is constructed by looking for regularities in data.

Example

This is an example to explain decision trees.

In this example we see the factors affecting sunburn in people who are coming to the beach on a sunny day.

The four attributes used in this case are, Hair, Height, Weight, Lotion, Result. These attributes are independent attributes or conditional attributes.

Reference: P. Winston, 1992.

Given Data

Independent Attributes / Condition Attributes

Dependent Attributes / Decision Attributes

Name	Hair	Height	Weight	Lotion	Result
Sarah	Blonde	average	Light	No	sunburned (positive)
Dana	Blonde	tall	Average	Yes	none (negative)
Alex	Brown	short	Average	Yes	None
Annie	Blonde	short	Average	No	Sunburned
Emily	Red	average	Heavy	No	Sunburned

Pete	Brown	Tall	Heavy	No	None
John	Brown	average	Heavy	No	None
Katie	Blonde	Short	Light	Yes	None

Phase 1: From Data to Tree

A) In this phase we understand the data and make a tree based on the data.

The initial step involves performing average entropy calculations on the complete data set for each of the four attributes:

Entropy Formula

Entropy, a measure from information theory, characterizes the (im) purity, or homogeneity, of an arbitrary collection of examples.

Given:

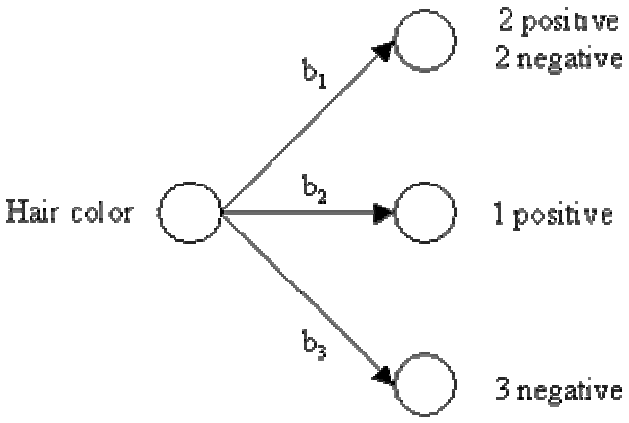
- n_b, the number of positive instances in branch b.
- n_{bc}, the total number of instances in branch b of class c.
- n_t, the total number of instances in all branches.

Entropy

- $$\text{Entropy} = \sum_c -\left(\frac{n_{bc}}{n_b}\right)\log_2\left(\frac{n_{bc}}{n_b}\right)$$
- As you move from perfect balance and perfect homogeneity, entropy varies smoothly between zero and one.
    - The entropy is zero when the set is perfectly homogeneous.
    - The entropy is one when the set is perfectly inhomogeneous.

Let's examine the entropy of attributes Hair Color, Height, Weight and Lotion.

Here we examine Haircolor first:



Attributes: Hair Color Reference: positive: sunburned

negative: none

b_1 = blonde Average Entropy = 0.50

b_2 = red

b_3 = brown

Sample average entropy calculation for the attribute "hair color"

$$= \sum_b \left((\frac{n_b}{n_t}) \times [\sum_c -(\frac{n_{bc}}{n_b}) \log_2 (\frac{n_{bc}}{n_b})] \right)$$

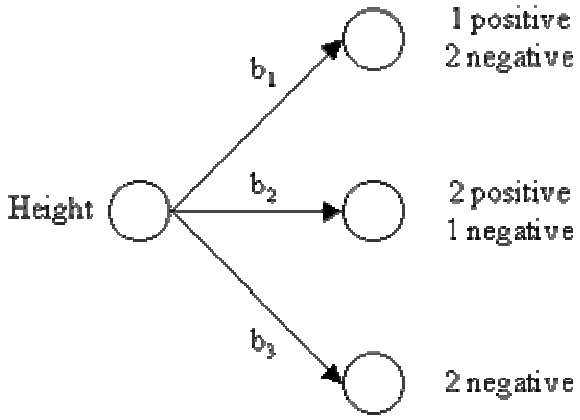
$$= \frac{4}{8} \times [-\frac{2}{4} \log_2 (\frac{2}{4}) - \frac{2}{4} \log_2 (\frac{2}{4})] + \frac{1}{8} \times (-\log_2 1) + \frac{3}{8} \times (-\log_2 1)$$

$$= \frac{4}{8} \times [-\frac{2}{4} \log_2 (\frac{2}{4}) - \frac{2}{4} \log_2 (\frac{2}{4})]$$

$$= 0\ 50$$

B) Now let's examine to see if the Height attribute has lower or higher entropy than Hair Color.

Entropy calculations for attribute "Height"



Attribute: Height Reference: positive: sunburned

negative: none

B_1 = short Average Entropy = 0.69

b_2 = average

b_3 = tall

Sample average entropy calculation for the attribute "Height"

$$= \frac{3}{8}*(-\frac{1}{3}\log_2(\frac{1}{3})-\frac{2}{3}\log_2(\frac{2}{3}))$$

$$+\frac{3}{8}(-\frac{2}{3}\log_2(\frac{2}{3})-\frac{1}{3}\log_2(\frac{1}{3}))+\frac{2}{8}(0)$$

$$=-\frac{1}{8}\log_2(\frac{1}{3})-\frac{1}{4}\log_2(\frac{2}{3})-\frac{1}{4}\log_2(\frac{2}{3})-\frac{1}{8}\log_2(\frac{1}{3})$$

$$=-\frac{1}{4}\log_2(\frac{1}{3})-\frac{1}{2}(\log_2(\frac{2}{3})$$

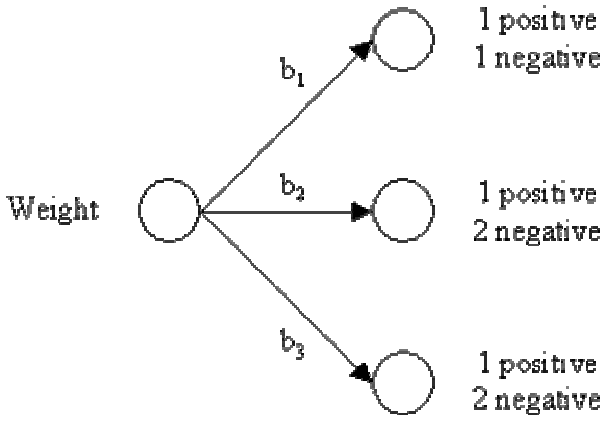
$$=-\frac{1}{4}\left(\frac{\log_{10}(\frac{1}{3})}{\log_{10}(2)}\right)-\frac{1}{2}\left(\frac{\log_{10}(\frac{2}{3})}{\log_{10}(2)}\right)$$

$$=0.3962+0.2925$$

$$=0.69$$

C) Now let's examine to see if the Weight attribute has lower or higher entropy than Hair Color, Height.

Entropy calculations for attribute "Weight"



Attribute: Weight Reference: positive: sunburned

negative: none

b_1 = light Average Entropy = 0.94

b_2 = average

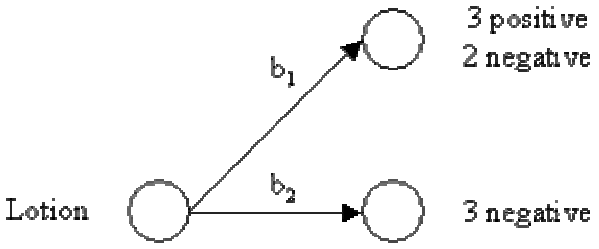
b_3 = heavy

Sample average entropy calculation for the attribute "Weight"

$$= \frac{2}{8}(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})) +$$
$$\frac{3}{8}(-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})) +$$
$$\frac{3}{8}(-\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})$$
$$= -\frac{1}{8}\log_2(\frac{1}{4}) - \frac{1}{8}\log_2(\frac{1}{3}) - \frac{1}{4}\log_2(\frac{2}{3}) - \frac{1}{8}\log_2(\frac{1}{3}) - \frac{1}{4}\log_2(\frac{2}{3})$$
$$= -\frac{1}{8}\log_2(\frac{1}{4}) - \frac{1}{4}\log_2(\frac{1}{3}) - \frac{1}{2}\log_2(\frac{2}{3})$$
$$= -\frac{1}{8}\left(\frac{\log_{10}(\frac{1}{4})}{\log_{10}2}\right) - \frac{1}{4}\left(\frac{\log_{10}(\frac{1}{3})}{\log_{10}2}\right) - \frac{1}{2}\left(\frac{\log_{10}(\frac{2}{3})}{\log_{10}2}\right)$$
$$= 0\ 25 + 0\ 3962 + 0.2925$$
$$= 0\ 94$$

D) Now let's examine to see if the Lotion attribute has lower or higher entropy than Hair Color, Height, and Weight

Entropy calculations for Attribute "Lotion"



Attribute: lotion Reference: positive: sunburned

negative: none

B_1 = no Average Entropy = 0.61

b_2 = yes

Sample average entropy calculation for the attribute" Lotion"

$$= \frac{5}{8}(-\frac{3}{5}\log_2(\frac{3}{5}) - \frac{2}{5}\log_2(\frac{2}{5})) + \frac{3}{8}(0)$$

$$= -\frac{3}{8}\log_2(\frac{3}{5}) - \frac{1}{4}\log_2(\frac{2}{5})$$

$$= -\frac{3}{8}\left(\frac{\log_{10}(\frac{3}{5})}{\log_{10} 2}\right) - \frac{1}{4}\left(\frac{\log_{10}(\frac{2}{5})}{\log_{10} 2}\right)$$

$$= 0.2764 + 0.3305$$

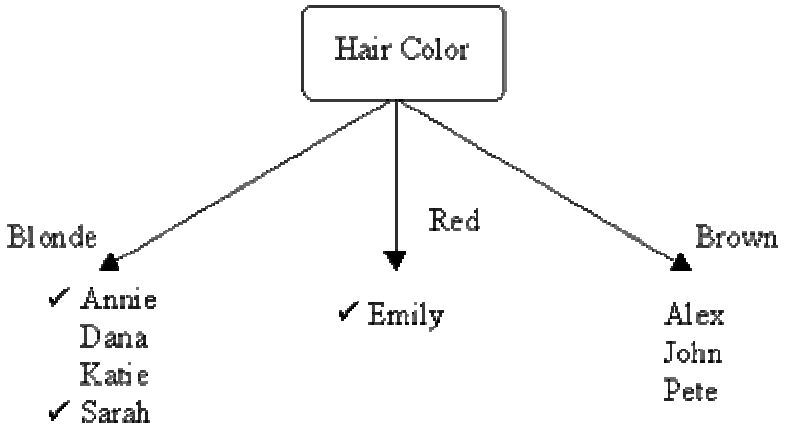
$$= 0.61$$

After the calculations, the attribute with least entropy is chosen.

Results

The attribute "hair color" is selected as the first test because it minimizes the entropy.

Attribute	Average Entropy
Hair Color	**0.50**
Height	0.69
Weight	0.94
Lotion	0.61



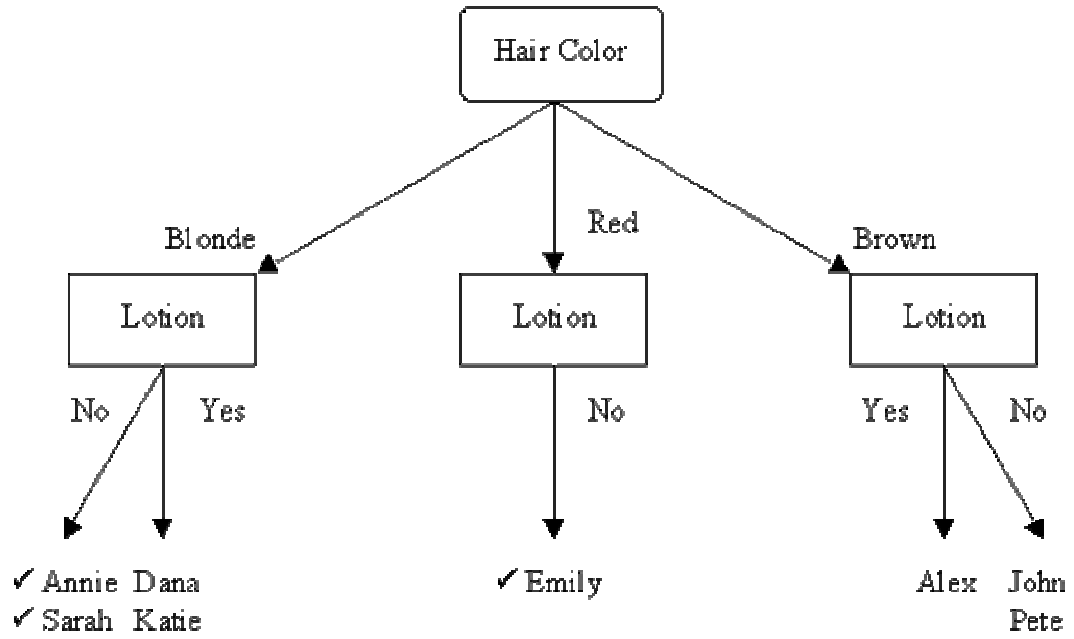
Similarly, we now choose another test to separate out the sunburned individuals from the blonde haired inhomogeneous subset, {Sarah, Dana, Annie, and Katie}.

Results

Attribute	Average Entropy
Height	0.50
Weight	1.00
Lotion	**0.00**

The attribute "lotion" is selected because it minimizes the entropy in the blonde hair subset.

Thus, using the "hair color" and "lotion" tests together ensures the proper identification of all the samples.



Rules:

Now we extract rules from decision trees from above.

Rule	If	Then
1	the person's hair color is blonde the person uses no lotion	The person gets sunburned.
2	the person's hair color is blonde the person uses lotion	Nothing happens.
3	the person's hair color is red	The person gets sunburned.
4	the person's hair color is brown	Nothing happens.

Eliminating the rules for getting the final rules:

Rule	If	Then
1	the person uses no lotion	The person gets sunburned.
2	the person uses lotion	Nothing happens.
3	the person's hair color is red	The person gets sunburned.
4	the person's hair color is brown	Nothing happens.

See5/C5.0

See5/C5.0 is a sophisticated tool to extract patterns from the data. C5.0 is a software extension of the basic ID3 algorithm designed by Quinlan. It can handle the over-fitting data, continuous attributes, handling the data with missing attribute values. It also largely improves the computational efficiency.

Data mining is all about extracting patterns from an organization's stored or warehoused data. These patterns can be used to gain insight into aspects of the organization's operations, and to predict outcomes for future situations as an aid to decision-making.

Patterns often concern the **categories** to which situations belong. For example, is a loan applicant creditworthy or not? Will a certain segment of the population ignore a mailout or respond to it? Will a process give high, medium, or low yield on a batch of raw material?

See5 (Windows 98/Me/2000/XP) and its Unix counterpart **C5.0** are sophisticated data mining tools for discovering patterns that delineate categories, assembling them into classifiers, and using them to make predictions.

Some important features:

- See5/C5.0 has been designed to analyze **substantial databases** containing thousands to hundreds of thousands of records and tens to hundreds of numeric or nominal fields.
- To maximize interpretability, See5/C5.0 classifiers are expressed as **decision trees** or **sets of if-then rules**, forms that are generally easier to understand than neural networks.
- See5/C5.0 is **easy to use** and does not presume advanced knowledge of Statistics or Machine Learning

Example

This is an illustration on how to use See5 for a medical application -- mining a database of thyroid assays to construct diagnostic rules for hypothyroidism. Each case concerns a single referral and contains information on the source of the referral, assays requested, patient data, and referring physician's comments. Here are three examples:

Attribute	Case 1	Case 2	Case 3	.....
Age	41	23	46	
sex	F	F	M	
on thyroxine	f	f	f	
query on thyroxine	f	f	f	
on antithyroid medication	f	f	f	
sick	f	f	f	
pregnant	f	f	not applicable	
thyroid surgery	f	f	f	
I131 treatment	f	f	f	
query hypothyroid	f	f	f	
query hyperthyroid	f	f	f	
lithium	f	f	f	
tumor	f	f	f	
goitre	f	f	f	
hypopituitary	f	f	f	
psych	f	f	f	
TSH	1.3	4.1	0.98	
T3	2.5	2	unknown	
TT4	125	102	109	
T4U	1.14	unknown	0.91	
FTI	109	unknown	unknown	
referral source	SVHC	other	other	
diagnosis	negative	negative	negative	
ID	3733	1442	2965	

This is exactly the sort of task for which See5 was designed. Each case belongs to one of a small number of mutually exclusive classes (negative, primary, secondary, compensated). Properties of

every case that *may* be relevant to its class are provided, although some cases may have unknown or non-applicable values for some attributes. There are 24 attributes in this example, but See5 can deal with any number of attributes.

See5's job is to find how to predict a case's class from the values of the other attributes. See5 does this by constructing a *classifier* that makes this prediction. As we will see, See5 can construct classifiers expressed as *decision trees* or as sets of *rules*.

Application filestem

Every See5 application has a short name called a filestem; we will use the filestem `hypothyroid` for this illustration. All files read or written by See5 for an application have names of the form filestem.extension, where filestem identifies the application and extension describes the contents of the file. The case of letters in both the filestem and extension is important -- file names `APP.DATA`, `app.data`, and `App.Data`, are all different. It is important that the extensions are written **exactly** as shown below, otherwise See5 will not recognize the files for your application.

Names file

Two files are essential for all See5 applications and there are three further optional files, each identified by its extension. The first essential file is the names file (e.g. `hypothyroid.names`) that describes the attributes and classes. There are two important subgroups of attributes:

- The value of an explicitly-defined attribute is given directly in the data. A *discrete* attribute has a value drawn from a set of nominal values, a *continuous* attribute has a numeric value, a *date* attribute holds a calendar date, a *time* attribute holds a clock time, a *timestamp* attribute holds a date and time, and a *label* attribute serves only to identify a particular case.

- The value of an implicitly-defined attribute is specified by a formula. (Most attributes are explicitly defined, so you may never need implicitly-defined attributes.)

The file `hypothyroid.names` looks like this:

```
diagnosis.                      | the target attribute

age:                            continuous.
sex:                            M, F.
on thyroxine:                   f, t.
query on thyroxine:             f, t.
on antithyroid medication:      f, t.
sick:                           f, t.
pregnant:                       f, t.
thyroid surgery:                f, t.
I131 treatment:                 f, t.
query hypothyroid:              f, t.
query hyperthyroid:             f, t.
lithium:                        f, t.
tumor:                          f, t.
goitre:                         f, t.
hypopituitary:                  f, t.
psych:                          f, t.
TSH:                            continuous.
T3:                             continuous.
TT4:                            continuous.
T4U:                            continuous.
FTI:=                           TT4 / T4U.
referral source:                WEST, STMW, SVHC, SVI, SVHD, other.

diagnosis:                    primary, compensated, secondary, negative.

ID:                             label.
```

What's in a name?

Names, labels, classes, and discrete values are represented by arbitrary strings of characters, with some fine print:

- Tabs and spaces are permitted inside a name or value, but See5 collapses every sequence of these characters to a single space.

- Special characters (comma, colon, period, vertical bar `|') can appear in names and values, but must be prefixed by the escape character `\'. For example, the name "Filch, Grabbit, and Co." would be written as `Filch\, Grabbit\, and Co\.'. (Colons in times and periods in numbers do not need to be escaped.)

Whitespace (blank lines, spaces, and tab characters) is ignored except inside a name or value and can be used to improve legibility. Unless it is escaped as above, the vertical bar `|' causes the remainder of the line to be ignored and is handy for including comments. This use of `|' should not occur inside a value.

The first line of the names file gives the classes, either by naming a discrete attribute (the *target* attribute) that contains the class value (as in this example), or by listing them explicitly. The attributes are then defined in the order that they will be given for each case.

Explicitly-defined attributes

The name of each explicitly-defined attribute is followed by a colon `:' and a description of the values taken by the attribute. There are six possibilities:

`continuous`

The attribute takes numeric values.

`date`

The attribute's values are dates in the form YYYY/MM/DD or YYYY-MM-DD, e.g. `1999/09/30` or `1999-09-30`.

`time`

The attribute's values are times in the form HH:MM:SS with values between `00:00:00` and `23:59:59`.

`timestamp`

The attribute's values are times in the form YYYY/MM/DD HH:MM:SS or YYYY-MM-DD HH:MM:SS, e.g. `1999-09-30 15:04:00`. (Note that there is a space separating the date and time.) a comma-separated list of names

The attribute takes discrete values, and these are the allowable values. The values may be prefaced by `[ordered]` to indicate that they are given in a meaningful ordering, otherwise they will be taken as unordered. For instance, the values `low, medium, high` are ordered, while `meat, poultry, fish, vegetables` are not. The former might be declared as

```
grade: [ordered] low, medium, high.
```

If the attribute values have a natural order, it is better to declare them as such so that See5 can exploit the ordering. (**NB:** The target attribute should not be declared as ordered.)

`discrete` N for some integer *N*

The attribute has discrete, unordered values, but the values are assembled from the data itself; *N* is the maximum number of such values. (This is not recommended, since the data

cannot be checked, but it can be handy for unordered discrete attributes with many values.)

(**NB:** This form cannot be used for the target attribute.)

`ignore`

The values of the attribute should be ignored.

`label`

This attribute contains an identifying label for each case, such as an account number or an order code. The value of the attribute is ignored when classifiers are constructed, but is used when referring to individual cases. A label attribute can make it easier to locate errors in the data and to cross-reference results to individual cases. If there are two or more label attributes, only the last is used.

Attributes defined by formulas

The name of each implicitly-defined attribute is followed by `:=' and then a formula defining the attribute value. The formula is written in the usual way, using parentheses where needed, and may refer to any attribute defined before this one. Constants in the formula can be numbers (written in decimal notation), dates, times, and discrete attribute values (enclosed in string quotes `""). The operators and functions that can be used in the formula are

- `+`, `-`, `*`, `/`, `%` (mod), `^` (meaning `raised to the power')
- `>`, `>=`, `<`, `<=`, `=`, `<>` or `!=` (both meaning `not equal')
- `and`, `or`
- `sin(...)`, `cos(...)`, `tan(...)`, `log(...)`, `exp(...)`, `int(...)` (meaning `integer part of')

The value of such an attribute is either continuous or true/false depending on the formula. For example, the attribute `FTI` above is continuous, since its value is obtained by dividing one number by another. The value of a hypothetical attribute such as

```
strange := referral source = "WEST" or age > 40.
```

would be either `t` or `f` since the value given by the formula is either true or false.

If the value of the formula cannot be determined for a particular case because one or more of the attributes appearing in the formula have unknown or non-applicable values, the value of the implicitly-defined attribute is unknown.

Dates, times, and timestamps

Dates are stored by See5 as the number of days since a particular starting point so some operations on dates make sense. Thus, if we have attributes

```
d1: date.
d2: date.
```

we could define

```
interval := d2 - d1.
gap := d1 <= d2 - 7.
d1-day-of-week := (d1 + 1) % 7 + 1.
```

`interval` then represents the number of days from `d1` to `d2` (non-inclusive) and `gap` would have a true/false value signaling whether `d1` is at least a week before `d2`. The last definition is a slightly non-obvious way of determining the day of the week on which `d1` falls, with values ranging from 1 (Monday) to 7 (Sunday).

Similarly, times are stored as the number of seconds since midnight. If the names file includes

```
start: time.
finish: time.
elapsed := finish - start.
```

the value of `elapsed` is the number of seconds from `start` to `finish`.

Timestamps are a little more complex. A timestamp is rounded to the nearest minute, but limitations on the precision of floating-point numbers mean that the values stored for timestamps from more than thirty years ago are approximate. If the names file includes

```
departure: timestamp.
arrival: timestamp.
flight time := arrival - departure.
```

the value of `flight time` is the number of minutes from `departure` to `arrival`.

Selecting the attributes that can appear in classifiers

An optional final entry in the names file affects the way that See5 constructs classifiers. This entry takes one of the forms

```
attributes included:
attributes excluded:
```

followed by a comma-separated list of attribute names. The first form restricts the attributes used in classifiers to those specifically named; the second form specifies that classifiers must not use any of the named attributes.

Excluding an attribute from classifiers is not the same as ignoring the attribute (see `ignore' above). As an example, suppose that numeric attributes `A` and `B` are defined in the data, but background knowledge suggests that only their difference is important. The names file might then contain the following entries:

```
. . .
A: continuous.
B: continuous.
Diff := A - B.
   . . .
attributes excluded: A, B.
```

In this example the attributes `A` and `B` could not be defined as `ignore` because the definition of `Diff` would then be invalid.

Data file

The second essential file, the application's data file (e.g. `hypothyroid.data`) provides information on the *training* cases from which See5 will extract patterns. The entry for each case consists of one or more lines that give the values for all explicitly-defined attributes. If the classes are listed in the first line of the names file, the attribute values are followed by the case's class value. Values are separated by commas and the entry is optionally terminated by a period. Once again, anything on a line after a vertical bar `|' is ignored. (If the information for a case occupies more than one line, make sure that the line breaks occur after commas.)

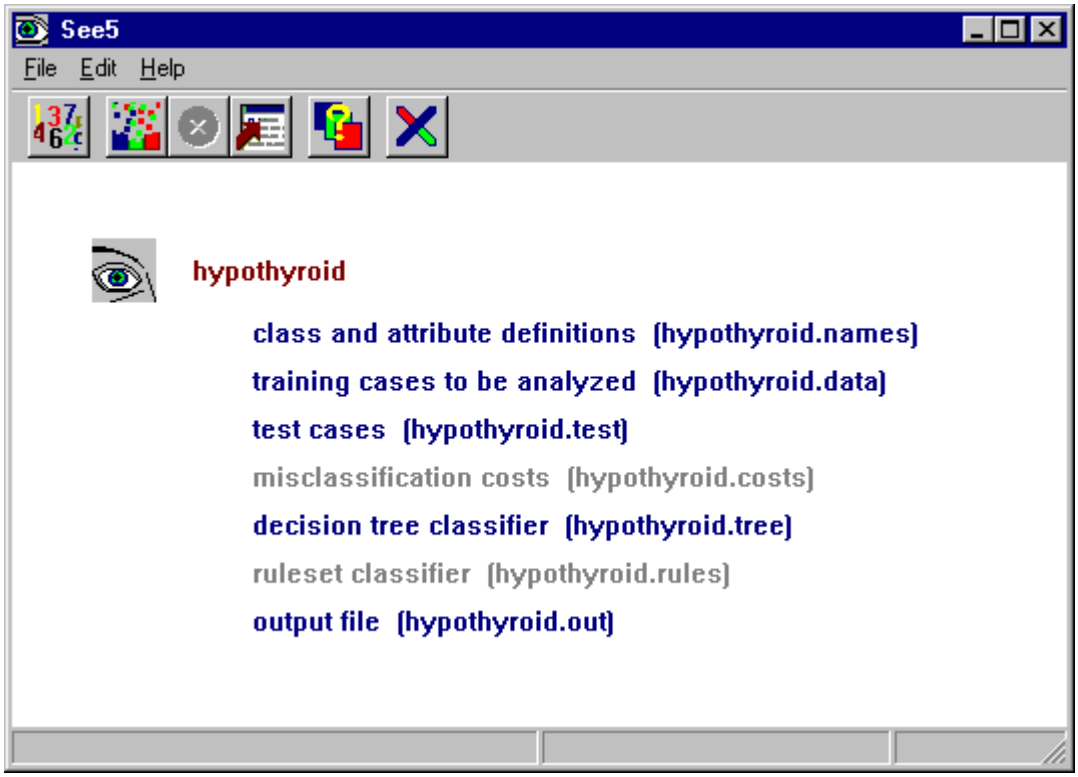
For example, the first three cases from file `hypothyroid.data` are:

```
41,F,f,f,f,f,f,f,f,f,f,f,f,f,f,f,1.3,2.5,125,1.14,SVHC,negative,3733
23,F,f,f,f,f,f,f,f,f,f,f,f,f,f,f,4.1,2,102,?,other,negative,1442
46,M,f,f,f,f,N/A,f,f,f,f,f,f,f,f,f,0.98,?,109,0.91,other,negative,2965
```

If there are no commas, then See5 will not be able to process the data. Notice that `?' is used to denote a value that is missing or unknown. Similarly, `N/A' denotes a value that is not applicable for a particular case. Also note that the cases do not contain values for the attribute `FTI` since its values are computed from other attribute values.

The third kind of file used by See5 consists of new test cases (e.g. `hypothyroid.test`) on which the classifier can be evaluated. This file is optional and, if used, has exactly the same format as the data file.

As a simple illustration, here is the main window of See5 after the hypothyroid application has been selected.



The main window of See5 has six buttons on its toolbar. From left to right, they are

Locate Data

invokes a browser to find the files for your application, or to change the current application;

Construct Classifier

selects the type of classifier to be constructed and sets other options;

Stop

interrupts the classifier-generating process;

Review Output

re-displays the output from the last classifier construction (if any);

Use Classifier

interactively applies the current classifier to one or more cases; and

Cross-Reference

shows how cases in training or test data relate to (parts of) a classifier and vice versa.

These functions can also be initiated from the File menu.

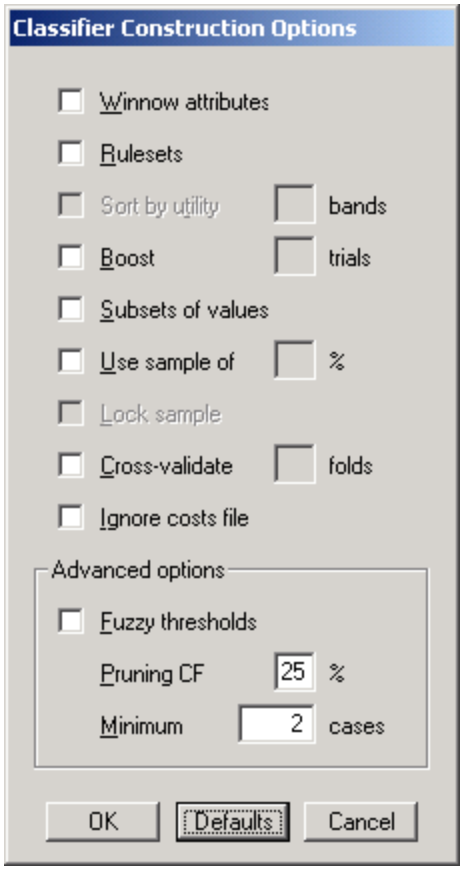
The Edit menu facilities changes to the names and costs files after an application's files have been located. On-line help is available through the Help menu.

Constructing Classifiers

Once the names, data, and optional files have been set up, everything is ready to use See5.

The first step is to locate the date using the Locate Data button on the toolbar (or the corresponding selection from the File menu). We will assume that the hypothyroid data above has been located in this manner.

There are several options that affect the type of classifier that See5 produces and the way that it is constructed. The Construct Classifier button on the toolbar (or selection from the File menu) displays a dialog box that sets out these classifier construction options:



Many of the options have default values that should be satisfactory for most applications.

Decision trees

When See5 is invoked with the default values of all options, it constructs a decision tree and generates output like this:

```
See5 [Release 1.16]     Tue Feb 19 09:04:01 2002

Class specified by attribute `diagnosis'

Read 2772 cases (24 attributes) from hypothyroid.data

Decision tree:

TSH <= 6: negative (2472/2)
TSH > 6:
:...FTI > 65:
    :...on thyroxine = t: negative (37.7)
    :   on thyroxine = f:
    :   :...thyroid surgery = t: negative (6.8)
    :       thyroid surgery = f:
    :       :...TT4 > 153: negative (6/0.1)
    :           TT4 <= 153:
    :           :...TT4 <= 37: primary (2.5/0.2)
    :               TT4 > 37: compensated (174.6/24.8)
    FTI <= 65:
    :...thyroid surgery = t:
        :...FTI <= 36.1: negative (2.1)
        :   FTI > 36.1: primary (2.1/0.1)
        thyroid surgery = f:
```

```
:...TT4 <= 61: primary (51/3.7)
    TT4 > 61:
    :...referral source in {WEST,SVHD}: primary (0)
        referral source = STMW: primary (0.1)
        referral source = SVHC: primary (1)
        referral source = SVI: primary (3.8/0.8)
        referral source = other:
        :...TSH > 22: primary (5.8/0.8)
            TSH <= 22:
            :...T3 <= 2.3: compensated (3.4/0.9)
                T3 > 2.3: negative (3/0.2)


Evaluation on training data (2772 cases):

          Decision Tree
        ----------------
        Size      Errors

          15    6( 0.2%)   <<


         (a)   (b)   (c)   (d)    <-classified as
        ----  ----  ----  ----
          60     3                (a): class primary
               154                (b): class compensated
                             2    (c): class secondary
                 1        2552    (d): class negative
```

```
Evaluation on test data (1000 cases):

          Decision Tree
        ----------------
        Size      Errors

          15    4( 0.4%)   <<


         (a)   (b)   (c)   (d)    <-classified as
        ----  ----  ----  ----
          31     1                 (a): class primary
           1    39                 (b): class compensated
                                   (c): class secondary
                 2         926     (d): class negative


Time: 0.1 secs
```

(Since hardware platforms can differ in floating point precision and rounding, the output that you see might not be exactly the same as the above.)

The first line identifies the version of See5 and the run date. See5 constructs a decision tree from the 2772 training cases in the file `hypothyroid.data`, and this appears next. Although it may not look much like a tree, this output can be paraphrased as:

```
if TSH is less than or equal to 6 then negative
else
if TSH is greater than 6 then
    if FTI is greater than 65 then
        if on thyroxine equals t then negative
        else
```

```
        if on thyroxine equals f then
            if thyroid surgery equals t then negative
            else
            if thyroid surgery equals f then
                if TT4 is greater than 153 then negative
                else
                if TT4 is less than or equal to 153 then
                    if TT4 is less than or equal to 37 then primary
                    else
                    if TT4 is greater than 37 then compensated
    else
    if FTI is less than or equal to 65 then
    . . . .
```

and so on. The tree employs a case's attribute values to map it to a *leaf* designating one of the classes. Every leaf of the tree is followed by a cryptic (*n*) or (*n*/*m*). For instance, the last leaf of the decision tree is `negative (3/0.2)`, for which *n* is 3 and *m* is 0.2. The value of *n* is the number of cases in the file `hypothyroid.data` that are mapped to this leaf, and *m* (if it appears) is the number of them that are classified incorrectly by the leaf.

Rulesets

Decision trees can sometimes be very difficult to understand. An important feature of See5 is its mechanism to convert trees into collections of rules called *rulesets*. The Rulesets option causes rules to be derived from trees produced as above, giving the following rules:

```
Rule 1: (31, lift 42.7)
        thyroid surgery = f
        TSH > 6
        TT4 <= 37
        ->  class primary  [0.970]
```

```
Rule 2: (23, lift 42.2)
        TSH > 6
        FTI <= 65
        referral source = SVI
        ->  class primary  [0.960]

Rule 3: (63/6, lift 39.3)
        TSH > 6
        FTI <= 65
        ->  class primary  [0.892]
```

Each rule consists of:

- A rule number -- this is quite arbitrary and serves only to identify the rule.
- Statistics `(`*n*`, lift` *x*`)` or `(`*n*`/`*m*`, lift` *x*`)` that summarize the performance of the rule. Similarly to a leaf, *n* is the number of training cases covered by the rule and *m*, if it appears, shows how many of them do not belong to the class predicted by the rule. The rule's accuracy is estimated by the Laplace ratio *(n-m+1)/(n+2)*. The lift *x* is the result of dividing the rule's estimated accuracy by the relative frequency of the predicted class in the training set.
- One or more conditions that must all be satisfied if the rule is to be applicable.
- A class predicted by the rule.
- A value between 0 and 1 that indicates the confidence with which this prediction is made.

The Problem

Server log files from CSIS website are used in this thesis. The server log files (412 MB) are stored in the hard drive. The log files are then queried to extract only the logs where the users used different keywords to reach the same web site.

For Example:

Keywords: pace, pace university, csis, graduate center of pace university, Westchester county courses

Search Engines: AltaVista, yahoo.com, google.com etc.

The Web page reached: / (csis.pace.edu)

The keywords and its relevance to the web site reached is understood manually by the human eye using knowledge and intuition.

When the user reaches the web site, the activity recorded tells if the user ever wanted to be on the site. Was he conducting any business or came to the site by mistake. These conclusions are drawn by understanding the log files of each and every particular user. The activity of each user in the main data files is again mined form the source file by using 'grep'.

The log file of the particular user, who has used a keyword in a search engine to reach this particular web site, is obtained. From this it is understood whether the user was serious in his search, or not.

The room for error in such a prediction exists. Never the less, it is more important to know the bigger picture, so that a specific description, idea is obtained form these server log files about large number of users.

As a result, the raw data about users, which is stored in server log files is mined to obtain knowledge so that, this information can be used to analyze the time and money spent on the web site is justified or not and to provide user friendly website. In a commercial way, this can be converted into retail dollars where the knowledge obtained can be used to make the web site more users friendly and give what they want.

The steps taken

a. Server log files

b.Cleaning the data

c. Specific user activity

d. Analysis of user activity

e. Creation of data sets for extracting decision trees using See5/C5.0

f. Understanding the decision trees

g. Analytics Derived

h. Applications

A: Server log files

The following are the server log files from the CSIS website. The following is a small part of the original log files. The original log file is 412MB and has 2.07 Million records. This is a huge file and patterns from this data are extracted.

format=%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%] "%Req->reqpb.clf-request%" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length% "%Req->headers.referer%" "%Req->headers.user-agent%" %Req->reqpb.method% %Req->reqpb.uri%

204.124.192.20 - - [02/Jul/2001:16:59:57 -0400] "GET /~knapp/is613.htm HTTP/1.0" 404 - "http://search.msn.com/spbasic.htm?MT=free%20relationship%20diagram%20software" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)" GET /~knapp/is613.htm

204.124.192.20 - - [02/Jul/2001:16:59:57 -0400] "GET /~knapp/pclc/csis-www.gif HTTP/1.0" 404 - "http://www.wol.pace.edu/~knapp/is613.htm" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)" GET /~knapp/pclc/csis-www.gif

172.17.82.108 - - [02/Jul/2001:16:59:48 -0400] "HEAD / HTTP/1.0" 200 - "-" "Ipswitch_WhatsUp/3.5" HEAD /

4.18.61.66 - - [02/Jul/2001:16:59:53 -0400] "GET /~bergin/iticse99/spreadsheet/sources/SpreadSheet.html HTTP/1.1" 404 - "http://www.google.com/search?q=Java+Spread+Sheet" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)" GET /~bergin/iticse99/spreadsheet/sources/SpreadSheet.html

4.18.61.66 - - [02/Jul/2001:16:59:54 -0400] "GET /~bergin/iticse99/spreadsheet/sources/pclc/csis-www.gif HTTP/1.1" 404 -

"http://www.csis.pace.edu/~bergin/iticse99/spreadsheet/sources/SpreadSheet.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)" GET /~bergin/iticse99/spreadsheet/sources/pclc/csis-www.gif

172.17.91.219 - - [02/Jul/2001:16:59:34 -0400] "HEAD / HTTP/1.0" 200 - "-" "Ipswitch_WhatsUp/3.5" HEAD /

172.17.82.108 - - [02/Jul/2001:16:59:33 -0400] "HEAD / HTTP/1.0" 200 - "-" "Ipswitch_WhatsUp/3.5" HEAD /

24.4.255.250 - - [02/Jul/2001:16:59:32 -0400] "GET /grendel/grehome.htm HTTP/1.0" 200 3053 "http://www.csis.pace.edu/grendel/" "Mozilla/4.0 (compatible; MSIE 5.0; Win3.1; ATHMWWW1.1;)" GET /grendel/grehome.htm

24.4.255.250 - - [02/Jul/2001:16:59:32 -0400] "GET /grendel/gawain.jpg HTTP/1.0" 200 84976 "http://www.csis.pace.edu/grendel/grehome.htm" "Mozilla/4.0 (compatible; MSIE 5.0; Win3.1; ATHMWWW1.1;)" GET /grendel/gawain.jpg

24.4.255.250 - - [02/Jul/2001:16:59:46 -0400] "GET /grendel/music.htm HTTP/1.0" 200 3399 "http://www.csis.pace.edu/grendel/gretoc.htm" "Mozilla/4.0 (compatible; MSIE 5.0; Win3.1; ATHMWWW1.1;)" GET /grendel/music.htm

24.4.255.251 - - [02/Jul/2001:16:59:31 -0400] "GET /grendel/ HTTP/1.0" 200 411 "http://search.yahoo.com/bin/search?p=Grendel" "Mozilla/4.0 (compatible; MSIE 5.0; Win3.1; ATHMWWW1.1;)" GET /grendel/

24.4.255.251 - - [02/Jul/2001:16:59:31 -0400] "GET /grendel/gretoc.htm HTTP/1.0" 200 966 "http://www.csis.pace.edu/grendel/" "Mozilla/4.0 (compatible; MSIE 5.0; Win3.1; ATHMWWW1.1;)" GET /grendel/gretoc.htm

24.4.255.251 - - [02/Jul/2001:16:59:31 -0400] "GET /grendel/sword2.gif HTTP/1.0" 200 6785 "http://www.csis.pace.edu/grendel/gretoc.htm" "Mozilla/4.0 (compatible; MSIE 5.0; Win3.1; ATHMWWW1.1;)" GET /grendel/sword2.gif

172.17.71.16 - - [02/Jul/2001:17:00:22 -0400] "HEAD / HTTP/1.0" 200 - "-" "Ipswitch_WhatsUp/3.5" HEAD /

195.168.68.73 - - [02/Jul/2001:17:00:29 -0400] "GET /newyorkgrandopera HTTP/1.0" 302 0 "http://www.goto.com/d/search/p/befree/?Promo=befree003830819131750094 92&Keywords=grand+opera+company&Promo=befree&Submit.x=5&Submit.y=9" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)" GET /newyorkgrandopera

195.168.68.73 - - [02/Jul/2001:17:00:29 -0400] "GET /newyorkgrandopera/ HTTP/1.0" 200 3693 "http://www.goto.com/d/search/p/befree/?Promo=befree003830819131750094 92&Keywords=grand+opera+company&Promo=befree&Submit.x=5&Submit.y=9" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)" GET /newyorkgrandopera/

B: Cleaning the data

These files are all the records of users who used different keywords but reached the same website.

Here we clean the vast data to obtain what we need from it. Using UNIX tools, the log files are cleaned to get the specific users activity from the server log files. We have 81614 records of the server log files where users have reached the same website with various keywords. In this thesis the point of interest is search strings (keywords), relevance and patterns from this information. Grep searches one or more input files for lines containing a match to a specified pattern. By default, grep prints the matching lines.

A *regular expression* is a pattern that describes a set of strings. Regular expressions are constructed analogously to arithmetic expressions, by using various operators to combine smaller expressions. `grep` understands two different versions of regular expression syntax: "basic"(BRE) and "extended"(ERE). In GNU `grep`, there is no difference in available functionality using either syntax. The room for error exists when the search engine parameters are found other lines.

The following is a small part of the original cleaned file.

12.22.22.2 / "http://enki.con.securecomputing.com/cgi-bin/tools/sf_frameset?url_id=849882&pre_cat=0&act=c&sf_url=http%3A%2F%2FCSFDESIGNS.COM&queue=c_New&url=http%3A%2F%2FCSFDESIGNS.COM&xinfo=&c_os=on&Memo=CLT+files+0417&framerows=274"

12.39.244.9 / "http://www.google.com/search?hl=en&safe=off&q=Pace+University+"

12.42.51.52 / "http://www.google.com/search?q=pace+university"

128.227.91.84 / "http://www.google.com/search?q=pace+university"

128.48.26.71 /

"http://www.google.com/u/CSUH?q=PACE&site=search&hl=en&safe=off&domains=csuhayward.edu&start=50&sa=N"

128.59.33.231 /

"http://www.google.com/search?q=Pace+University&btnG=Google%B7j%B4M&hl=zh-TW&lr="

12.88.114.166 / "http://search.lycos.com/main/?query=pace+graduate+center&rd=y"

12.89.147.248 / "http://www.worldnet.att.net/cgi-bin/websearch?cmd=qry&qry=health+curriculums"

12.89.171.207 / "http://www.google.com/search?q=Pace+University"

129.239.3.227 /

"http://search.excite.com/search.gw?search=%2BRAM+%2Bmemory+%2Bwaveform+%2Bexplain+%2B%22set+up+and+hold%22+site:www.csis.pace.edu"

129.42.208.140 / "http://www.google.com/search?q=pace+university&chk=on"

12.98.161.42 / "http://www.google.com/search?hl=en&safe=off&q=Pace+University"

130.237.68.55 /

"http://www.altavista.com/sites/search/web?q=%22Pace+University%22&pg=q&kl=XX&search=Search"

132.236.140.23 / "http://www.altavista.com/cgi-bin/query?pg=n200&stype=stext&user=msn&poa=msn&q=Pace%20University&stq=0"

132.236.140.23 / "http://www.altavista.com/cgi-bin/query?pg=n200&stype=stext&user=msn&poa=msn&q=Pace%20University&stq=20"

132.238.19.252 / "http://www.google.com/search?q=pace+university+"

134.129.186.95 /

"http://www.altavista.com/sites/search/web?q=pace+university&pg=q&kl=XX&search=Search"

134.65.2.66 /

http://www.google.com/search?hl=en&safe=off&q=pace+university+westchester&btnG=Google+Search

196.2.44.152 /~bergin/

"http://google.yahoo.com/bin/query?p=What+are+design+patterns+in+Object+Orientation%3f&b=21&hc=0&hs=0"

196.42.35.25 /~bergin/ "http://google.yahoo.com/bin/query?p=Joseph+Bergin&hc=0&hs=0"

196.42.47.182 /~bergin/ "http://google.yahoo.com/bin/query?p=Joseph+Bergin&hc=0&hs=0"

198.102.62.250 /~bergin/

"http://www.google.com/search?q=professor+home+page&hl=en&safe=off&start=20&sa=N"

198.146.124.5 /~bergin/

"http://search.excite.com/search.gw?c=web&search=Hildegard+in+Bergin&onload="

198.234.216.212 /~bergin/

"http://us.f115.mail.yahoo.com/ym/ShowLetter?MsgId=6097_4720817_1824_887_2767_0_0&YY=69180&inc=10&order=down&sort=date&pos=1&box=Inbox"

199.203.97.113 /~bergin/

"http://www.alltheweb.com/search?cat=web&advanced=1&type=all&query=OOP+C%2B%2B&jsact=&lang=any&charset=utf-8&wf%5Bn%5D=3&wf%5B0%5D%5Br%5D=&wf%5B0%5D%5Bq%5D=&wf%5B0%5D%5B

w%5D=&wf%5B1%5D%5Br%5D=%2B&wf%5B1%5D%5Bq%5D=&wf%5B1%5D%5Bw%5D=&wf%5B2%5D%5Br%5D=-&wf%5B2%5D%5Bq%5D=&wf%5B2%5D%5Bw%5D=&dincl=edu&dexcl=&hits=10&nooc=on"

199.240.129.203 /~bergin/ "http://www.google.com/search?q=C%2B%2B+teaching+examples"

199.67.138.20 /~bergin/ "http://google.yahoo.com/bin/query?p=Joseph+Bergin&hc=0&hs=0"

199.67.138.20 /~bergin/ "http://google.yahoo.com/bin/query?p=Joseph+Bergin&hc=0&hs=0"

200.205.16.130 /~bergin/ "http://search.msn.com.br/spbasic.htm?MT=karel&UDo=br&LA=portuguese&CO=10&UN=doc"

200.29.159.140 /~bergin/ "http://www.google.com/search?q=data+passing+to+web+page"

200.61.80.18 /~bergin/ http://www.google.com/search?hl=en&safe=off&q=bergin

C: Specific user activity from the test cases

After compiling the information about the users who have come to the same website using different keywords, the example to be worked upon is chose from the new log file. Here we study the specific user activity on the website using “GREP”

The following is a partial list of particular user activity for a specific user:

200.205.16.130 - - [05/Jul/2001:19:55:49 -0400] "GET /~bergin/indexbg.gif HTTP/1.1" 200 1998 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/indexbg.gif

200.205.16.130 - - [05/Jul/2001:19:55:50 -0400] "GET /~bergin/bergin3.gif HTTP/1.1" 200 1447 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/bergin3.gif

200.205.16.130 - - [05/Jul/2001:19:55:51 -0400] "GET /~bergin/greybull.gif HTTP/1.1" 200 1022 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/greybull.gif

200.205.16.130 - - [05/Jul/2001:19:55:52 -0400] "GET /~bergin/spicy.gif HTTP/1.1" 200 145 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/spicy.gif

200.205.16.130 - - [05/Jul/2001:19:55:54 -0400] "GET /~bergin/pix/key1.gif HTTP/1.1" 200 850 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/pix/key1.gif

200.205.16.130 - - [05/Jul/2001:19:55:44 -0400] "GET /~Bergin HTTP/1.1" 302 0 "http://search.msn.com.br/spbasic.htm?MT=karel&UDo=br&LA=portuguese&CO=10&UN=doc" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin

200.205.16.130 - - [05/Jul/2001:19:55:46 -0400] "GET /~bergin/ HTTP/1.1" 200 17956 "http://search.msn.com.br/spbasic.htm?MT=karel&UDo=br&LA=portuguese&CO=10&UN=doc" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/

200.205.16.130 - - [05/Jul/2001:19:55:51 -0400] "GET /~bergin/linesep.GIF HTTP/1.1" 200 1380 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/linesep.GIF

200.205.16.130 - - [05/Jul/2001:19:55:52 -0400] "GET /~bergin/new2.gif HTTP/1.1" 200 682 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/new2.gif

200.205.16.130 - - [05/Jul/2001:19:55:53 -0400] "GET /~bergin/new.gif HTTP/1.1" 200 688 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/new.gif

200.205.16.130 - - [05/Jul/2001:19:55:54 -0400] "GET /~bergin/macmade.gif HTTP/1.1" 200 873 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/macmade.gif

200.205.16.130 - - [05/Jul/2001:19:56:17 -0400] "GET /~bergin/karel.html HTTP/1.1" 200 5302 "http://www.wol.pace.edu/~bergin/" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/karel.html

200.205.16.130 - - [05/Jul/2001:19:56:31 -0400] "GET /~bergin/KWorld/karelwin.html HTTP/1.1" 200 17792 "http://www.wol.pace.edu/~bergin/karel.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/KWorld/karelwin.html

200.205.16.130 - - [05/Jul/2001:19:56:26 -0400] "GET /~bergin/Karel.jpg HTTP/1.1" 200 44113 "http://www.wol.pace.edu/~bergin/karel.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/Karel.jpg

200.205.16.130 - - [05/Jul/2001:19:57:01 -0400] "GET /~bergin/KWorld/kwin6.gif HTTP/1.1" 200 1803 "http://www.wol.pace.edu/~bergin/KWorld/karelwin.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/KWorld/kwin6.gif

200.205.16.130 - - [05/Jul/2001:19:57:03 -0400] "GET /~bergin/KWorld/kwin8.gif HTTP/1.1" 200 3515 "http://www.wol.pace.edu/~bergin/KWorld/karelwin.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/KWorld/kwin8.gif

200.205.16.130 - - [05/Jul/2001:19:56:37 -0400] "GET /~bergin/KWorld/kwin1.gif HTTP/1.1" 200 18526 "http://www.wol.pace.edu/~bergin/KWorld/karelwin.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/KWorld/kwin1.gif

200.205.16.130 - - [05/Jul/2001:19:56:48 -0400] "GET /~bergin/KWorld/kwin2.gif HTTP/1.1" 200 7054 "http://www.wol.pace.edu/~bergin/KWorld/karelwin.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/KWorld/kwin2.gif

200.205.16.130 - - [05/Jul/2001:19:56:54 -0400] "GET /~bergin/KWorld/kwin3.gif HTTP/1.1" 200 4783 "http://www.wol.pace.edu/~bergin/KWorld/karelwin.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/KWorld/kwin3.gif

200.205.16.130 - - [05/Jul/2001:19:56:56 -0400] "GET /~bergin/KWorld/kwin4.gif HTTP/1.1" 200 1769 "http://www.wol.pace.edu/~bergin/KWorld/karelwin.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/KWorld/kwin4.gif

200.205.16.130 - - [05/Jul/2001:19:56:59 -0400] "GET /~bergin/KWorld/kwin5.gif HTTP/1.1" 200 7133 "http://www.wol.pace.edu/~bergin/KWorld/karelwin.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/KWorld/kwin5.gif

200.205.16.130 - - [05/Jul/2001:19:57:01 -0400] "GET /~bergin/KWorld/kwin7.gif HTTP/1.1" 200 1821 "http://www.wol.pace.edu/~bergin/KWorld/karelwin.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/KWorld/kwin7.gif

200.205.16.130 - - [05/Jul/2001:19:57:16 -0400] "GET /~bergin/temp/findkarel.html HTTP/1.1" 200 5375 "http://www.wol.pace.edu/~bergin/karel.html" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)" GET /~bergin/temp/findkarel.html

200.205.16.130 - - [03/Aug/2001:17:38:58 -0400] "GET /~ryeneck/mahony/LMPROJ8.HTM HTTP/1.1" 404 332 "http://www.geometria.com.br/database.asp?codigo=5" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)" GET /~ryeneck/mahony/LMPROJ8.HTM

D: **Analysis of the user activity:**

The user keyword is known from the log files. If the keyword matches the information based on the page and there is considerable time spent, both by clicking or surfing (reading on) then it is understood as a serious user. The number of clicks (HTML pages visited) are counted, amount of time spend is also known. All these factors are taken into account and understood whether there was a keyword match, serious user or not. This is again tabulated to extract decision trees using attributes like keyword, destination (yes/no).

E: **Creation of data sets for See5/C5.0**

See5/C5.0 is used to extract patterns from the user behavior on the website. The patterns are extracted with website destination as key attribute. Keyword as attribute is also used.

In this case, the analysis has been for all the users who have come to the root page of the csis. If the user is exiting the page immediately by not accessing any further information in just few seconds (in most cases its 2 sec, but the duration has varied from 1-4 secs), then the users destination has not been CSIS.

Stages are nothing but the pages accessed by the user right immediately after the query in the search website. If the user just visits the root page then its just stage 1. If more pages have been visited by the user then accordingly stages have been given numbers. Stage 3 is given either number 1 or more to indicate the number of pages, he/she went to.

1. Data File for See5/C5.0:

The below csis.data file is a partial data file from the original data file.

csis.data

pace university,1,1,1,yes.

pace university,1,0,0,no.

PACE,1,0,0,no.

Pace University,1,0,0,no.

health curriculums,1,0,0,no.

Pace University,1,0,0,no.

Pace University,1,0,0,no.

Pace University,1,0,0,no.

Pace University,1,0,0,no.

Pace University,1,0,0,no.

pace university,1,0,0,no.

pace university,1,0,0,no.

Pace University,1,1,0,yes.

PACE,1,0,0,no.

pace university,1,0,0,no.

pace university,1,0,0,no.

Health Curriculum,1,0,0,no.

pace university,1,0,0,no.

pace university,1,0,0,no.

New Rochelle NY,1,0,0,no.

pace university,1,1,0,yes.

computer couse new york city,1,1,1,yes.

PACE UNIVERSITY,1,1,6,yes.

pace university,1,0,0,no.

pace university,1,1,0,yes.

pace,1,1,2,yes.

pace,1,1,0,yes.

pace university,1,0,0,no.

Pace University Westchester,1,0,0,no.

pace university,1,1,9,yes.

pace university ny,1,1,9,yes.

UNIVERSITY NEW ROCHELLE,1,1,2,yes.

pace university,1,1,2,yes.

pace university,1,0,0,no.

pace university,1,1,15,yes.

csis pace edu pclc unixc outline html,1,0,0,yes.

pace university,1,1,3,yes.

PAce University,1,0,0,no.

PAce University,1,0,0,no.

Pace University,1,0,0,no.

pace university,1,0,0,no.

www csis pace edu grendel prjs3c analysis,1,1,2,yes.

csis\.pace.edu,1,1,3,yes.

pace university,1,0,0,no.

pace university,1,0,0,no.

pace university,1,0,0,no.

paceuniversity,1,1,15,yes.

Pace University,1,0,0,no.

Pace University Pleasantville,1,0,0,no.

health curriculms,1,1,1,yes.

cis pace edu,1,0,0,no.

pace university,1,1,5,yes.

new rochelle university,1,0,0,no.

pace university,1,0,0,no.

pace university,1,0,0,no.

VERDI REUNIFICATION site www\.csis\.pace\.edu,1,1,0,yes.

Pace University,1,0,0,no.

csis pace edu anderson cis101 lab,1,0,0,no.

pace university,1,0,0,no.

PACE,1,0,0,no.

pace university,1,1,0,yes.

pace university,1,1,7,yes.

pace university,1,1,4,yes.

2. Names file for See5/C5.0

The below csis.names file is a partial data file from the original data file.

csis.names:

csis destination. | The target attribute

keyword: pace university, PACE, Pace University,health curriculums,Health Curriculum,New Rochelle NY,SUNY+Pace,westchester county center,pace,pace university graduate' ,pace university westchester,ol baker edu,ancient writings and alphabet,sexual health curriculum,PACE UNIVERSITY,PACE UNIVERSITY,pace university graduate school,pace graduate center, ram memory wave form explain set up and hold www.csis.pace.edu,westchester payroll courses,pace university information systems,big-omega already sorted insertion,MVC pattern,computer couse new york city,new rochelle university,www csis pace edu grendel prjs3c analysis,csis.pace.edu,UNIVERSITY NEW ROCHELLE,pace University,PACE INIVERSITY,Pace University Westchester,pace university ny,pace university graduate,UNIVERSITY NEW ROCHELLE,csis pace edu pclc unixc outline html,PAce University,paceuniversity,Pace University Pleasantville,health curriculms,cis pace edu,VERDI REUNIFICATION site www.csis.pace.edu,csis pace edu anderson cis101 lab.

stg1: continuous.

stg2: continuous.

stg3: continuous.

tstg:= stg1+stg2+stg3.

csis destination: yes,no.

F: Understanding the decision trees:

```
See5 [Release 1.15]     Wed Sep 25 03:13:17 2002
Class specified by attribute `csis destination'
Read 120 cases (6 attributes) from csis.data
Decision tree:
stg2 <= 0: no (85/3)
stg2 > 0: yes (35)

Evaluation on training data (120 cases):
           Decision Tree
         ----------------
         Size      Errors
            2    3( 2.5%)   <<

          (a)   (b)    <-classified as
         ----  ----
           35     3    (a): class yes
                 82    (b): class no

Time: 0.2 secs

See5 [Release 1.15]     Wed Sep 25 03:20:38 2002
    Options:
        Rule-based classifiers
        Focus on errors (ignore costs file)
Class specified by attribute `csis destination'
Read 120 cases (6 attributes) from csis.data
Rules:
Rule 1: (35, lift 3.1)
        stg2 > 0
        ->  class yes  [0.973]
Rule 2: (85/3, lift 1.4)
        stg2 <= 0
        ->  class no  [0.954]
Default class: no

Evaluation on training data (120 cases):
                Rules
```

```
----------------
  No      Errors

   2    3( 2.5%)   <<


 (a)   (b)    <-classified as
----  ----
  35     3    (a): class yes
         82    (b): class no

Time: 0.0 secs
```

Explanation:

The tree employs a case's attribute values to map it to a *leaf* designating one of the classes. Every leaf of the tree is followed by a cryptic (*n*) or (*n*/*m*).

The value of *n* is the number of cases in the file `csis.data` that are mapped to this leaf, and *m* (if it appears) is the number of them that are classified incorrectly by the leaf.

For Rule#1:

n=35, these are the number of people who went two atleast two pages after coming to the web site and their destination was this site.

The rule's accuracy is estimated by the Laplace ratio *(n-m+1)/(n+2).*=(35-0+1)/(35+2)=0.973

The lift *x* is the result of dividing the rule's estimated accuracy by the relative frequency of the predicted class in the training set.

lift x=0.973/(35/120)= 3.1

For Rule#2:

n=85, m=3, n-m=82

82 are the number of people who came to this site through a search engine, and their destination wasn't this site. They left the site in an average of less than 2-3 seconds.

Only 3 cases spent more time on the site with out clicking further. So these three cases are not classified by this rule.

As a result these rule sets give an insight into how people behave on the web site. This information can be used to address the problems associated with web and making it more interactive.

The analysis of users tells what their choice has been after coming to the site. As a result we can use this information for making a more user friendly site so that the web experience is improved and goals are achieved.

G: Analytics Derived

Most people know in less than three seconds (<2-3 Secs) if they came to the right website by searching. This is shown by recorded server log files. The way people communicate with information is superficial / abstract way rather than with inner details. If the user wanted to know about CSIS or any particular information about course, they have used the keyword pace university (ignoring the case) more often to reach CSIS site than use the specific subject in CSIS as a search string. This has been proved for most of the examples worked upon.

Analytics based on Keywords:

Sl. No	Website	Keywords	Unusual Keywords	Total No of Users
1	http://csis.pace.edu	1.Combination of Pace, pace university	1.UNIVERSITY NEW ROCHELLE	120

		No of users: 103 2.Pace university No of users: 90	2.sexual health curriculum	
2	http://csis.pace.edu/~benjamin	1. Combination of Paul Benjamin No of users: 2 2. Paul Benjamin No of users: 10	1. Paul Benjamin office hours	10
3	http://csis.pace.edu/~bergin	1. Combinations of Bergin No of users: 81 2. Combinations of Karel No of users: 17 3. Combinations of patterns No of users: 14	1. abbey 2. key image	156
4.	http://csis.pace.edu/pclc	1.Combinations of pace No of users: 45 2. pace university No of users: 32 3. Combinations of computer learning center No of Users: 19	1. Cheltenham computer training 2. Microsoft logo gear	122

Note: Ignore the case for the keywords

Majority of the users after coming to the site have again searched for the specific information by clicking two or three links. The returned pages may contain the keyword as part of information in the page or as another hyper link. As a result the user again searches on the returned page for the information based on his/ her interests. Hence, the returned page which is just a static page with keyword info is again leading the user to look further for the final destination.

Hence it's become more like searching for information more than once for the same subject. If the returned page is of more than two "screen shots" then the user's scrolls up and down for information. As a result more time is taking for the user to reach his goal.

We have 145 Million people surfing internet in US today. If they waste on an average 1 sec on the search returned page to look for what they have been searching for, then we are wasting 145 Million seconds. If this time can be put into productive work or multiplied by some $ figure, then the wasted value is staggering.

Is there a way where we can address this issue from both the web surfer point of view and the site administrator point of view?

H: Applications

A Real Life Scenario

Kat walks into CVS Pharmacy with the intention to buy chocolates. She has chocolates in her mind and walks into the store. At the entrance a sales clerk greets her and says, "Madam, we have Mars, M&M's, Snickers, Reeses". Which of these do you want? Please feel free to go ahead and look around, if any of these are not in your mind. Kat chooses M&M's, pays at the counter and walks out. This is a very lovely situation where Kat didn't have to walk into the lengthy aisles, search for the chocolates again there. She decides on M&M's, picks them up, pays for it and leaves.

But how probable is this where the store clerk is right at the door, greeting with what you need? It's really hard to believe if we can ever have such a convenience. But with the knowledge we know about the web site users, their search keywords, the pages they surfed, can a possible "Kat in CVS Pharmacy Store" scenario be created?

A Real Life Scenario in Cyberspace:

A web application has been developed which can make the searching for information on a web page easy.

This is a new web application and it hasn’t been used on any web site as of now. As a result, it’s new, innovative, makes life easier.

Personalization for the user at the website end based on his search keywords.

Various options have been looked into to make the website more user and search friendly.

Applications:

1. **Pop-Up**

When the user comes to the website via a search engine, we can create a java script pop-up, which will load automatically when the web page is returned to the user. The pop-up will display the links based on the search query used by the user.

Advantages:

1. JavaScript pop-ups, effects are much faster to download than some other front-end technologies like Flash and Java applets.
2. Users need to download a plug-in before they can view the JavaScript
3. The user doesn't need any extra tools to write JavaScript, any plain text or HTML editor is enough.

Disadvantages:

1. As with most web development, it is entirely related to browser compatibility.
2. But with the advent of anti pop up software it is very unlikely that the web surfer will take a look at it and click on the link.
3. Pop ups will not really drive home the point because the surfer will be looking at the web page returned by the search engine rather than pay attention to a pop up which will make life easy.

2. **DHTML**

DHTM can be used to create links based on search query on the web page and load them with the web page.

Advantages:

1. Since it is a combination of CSS and JavaScript, DHTML can be created easily.

Disadvantages:

1. DHTML only works on higher level browsers (Netscape 4 and up, Internet Explorer 4 and up).
2. Netscape is very unstable when running DHTML.
3. DHTML is tricky - it takes a lot of practice to learn how to do it right.

3. Web Application (Java Servlet)

Explanation of Working:

With the knowledge of the http referrer info, the search query in the http referrer is used. The most common links for that specific search query are displayed right on top of the page. As a result links pertaining to the keyword are displayed dynamically at the top of the page. The links are not default links for every search query, but will change based on the query string.

The links will be displayed in the page as the page loads for the user. This will let the user click on the respective link immediately based on his needs or can look around the page. As a result time is saved for the user and efficiency is improved at both the ends. If combinations of keywords in the search query are used then the links based on that combination are displayed without compromising the quality of the interactivity.

Web Server used on the local host machine is: Apache Web Server

Advantages:

Web Application (Servlet) (Used in the demo on a local machine)

1. Interactive experience: Servlets offer interactive experience. They can generate dynamic content and many servlets can be coupled together to offer interactivity to the web applications. All this can be achieved at the same time while it is secure. The servlets can handle errors safely due to java's exception handling mechanisms.

2. Portability: The servlets are portable like Java. The servlets can be run on any java enabled web server without any code changes.

3. Powerful: Servlets can talk directly to the web servers. Multiple servlets can share data.

4. Efficiency and Endurance: For each browser request, the servlet spawns a light weight thread. This is faster and more efficient than spawning a new operating system process. Servlets in general are enduring objects. Because a servlets stays in the servers memory as a single object instance, it automatically maintains its state and can hold on to external resources like databases.

5. Extensibility and Flexibility: The servlet API is extensible. The API includes classes that are optimized for HTTP servlets. Servlets are also quite flexible.

Disadvantages

1. Developers will need to learn Java

2. Web Administrator will need to learn how to install and maintain Java Servlets

For Example:

A) Website: http://csis.pace.edu/~bergin

Case 1: When the user comes to the website directly by typing the address in the browser

When the http referrer info is null, then the default links based on the webpage are displayed.

Case 2:

Keyword: karel by bergin

As the page loads links based on the http referrer information (keyword) are displayed in the top of the page.

Case 3:

Keyword: pedagogical patterns

As the page loads links based on the http referrer information (keyword) are displayed in the top of the page.

Case 4:

Keyword: patterns and karel

As the page loads links based on the http referrer information (keyword) are displayed in the top of the page.

B) Website:

http://csis.pace.edu/~benjamin/

http Referrer: Null

When the http referrer is null, then the default links based on the webpage are displayed.

Case 2:

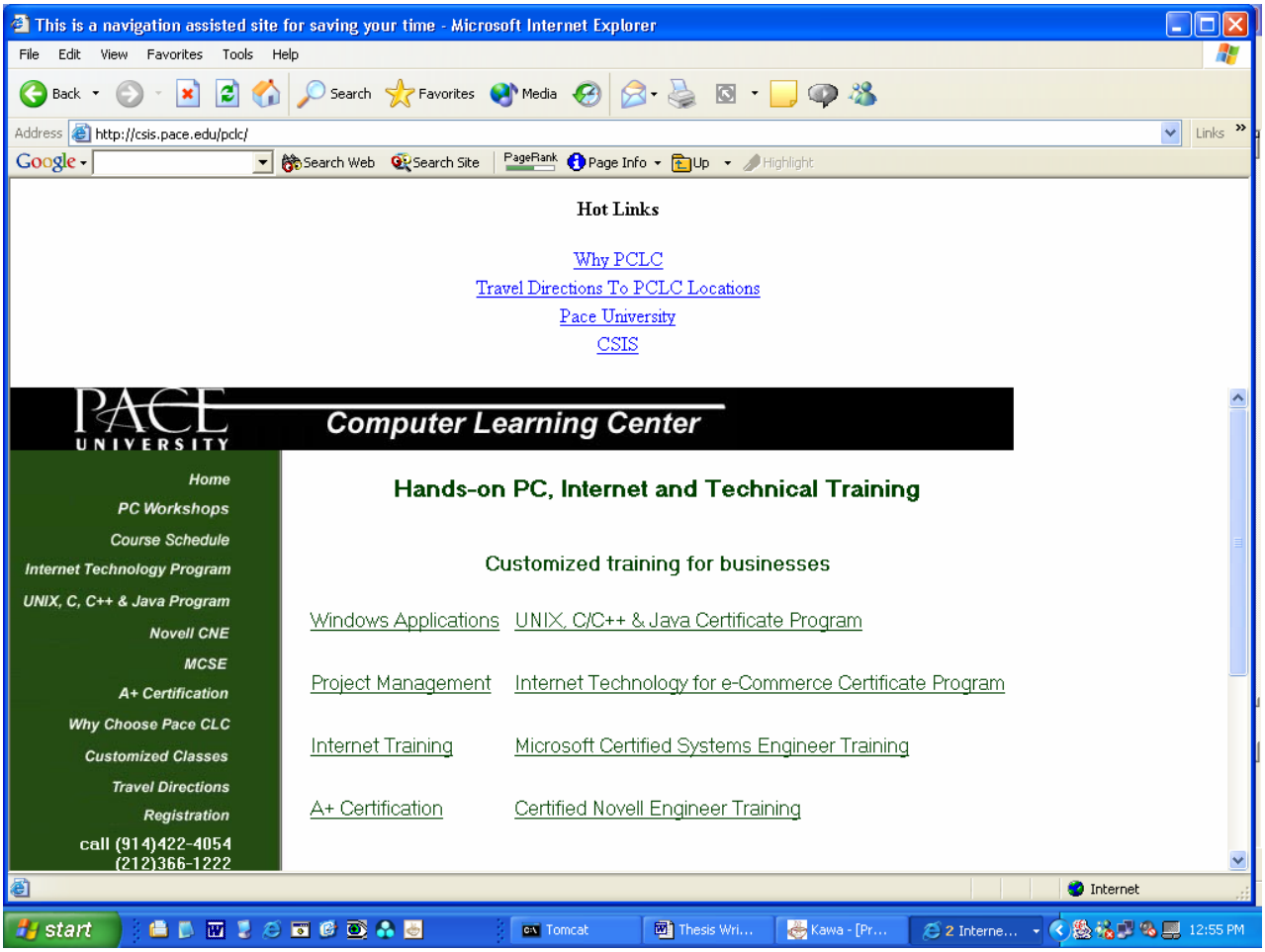
Keyword: paul benjamin

As the page loads links based on the http referrer information (keyword) are displayed in the top of the page.

C) Website:

http://csis.pace.edu/pclc/

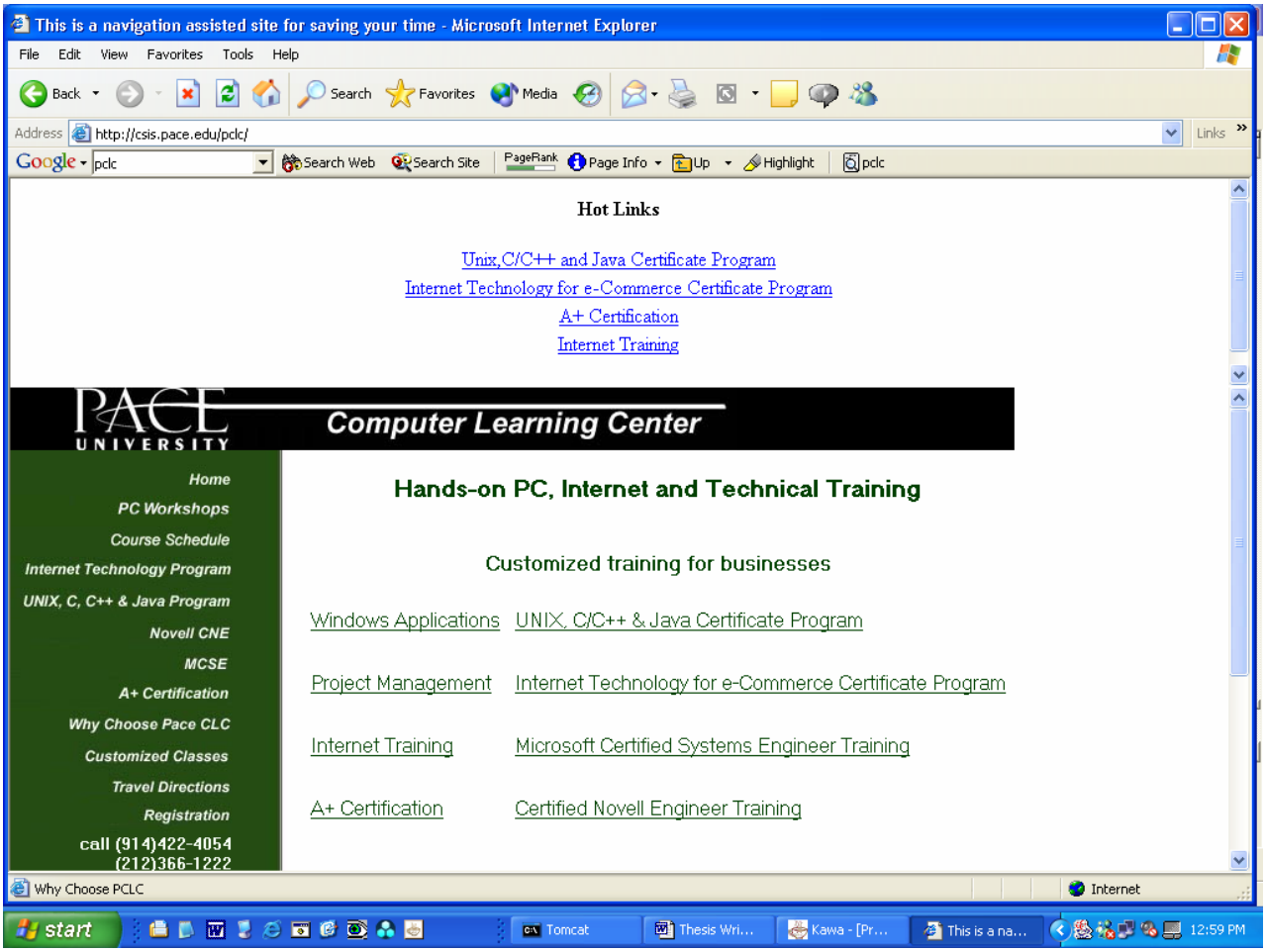
Case 1: When the user came to the website directly



When the http referrer info is null, then the default links based on the webpage are displayed.

Case 2:

Keyword: pclc



As the page loads links based on the http referrer information (keyword) are displayed in the top of the page.

Case 3:

Keyword: microsoft certificate courses in pclc

As the page loads links based on the http referrer information (keyword) are displayed in the top of the page.

Case 4:

Keyword: certificate programs in pace university

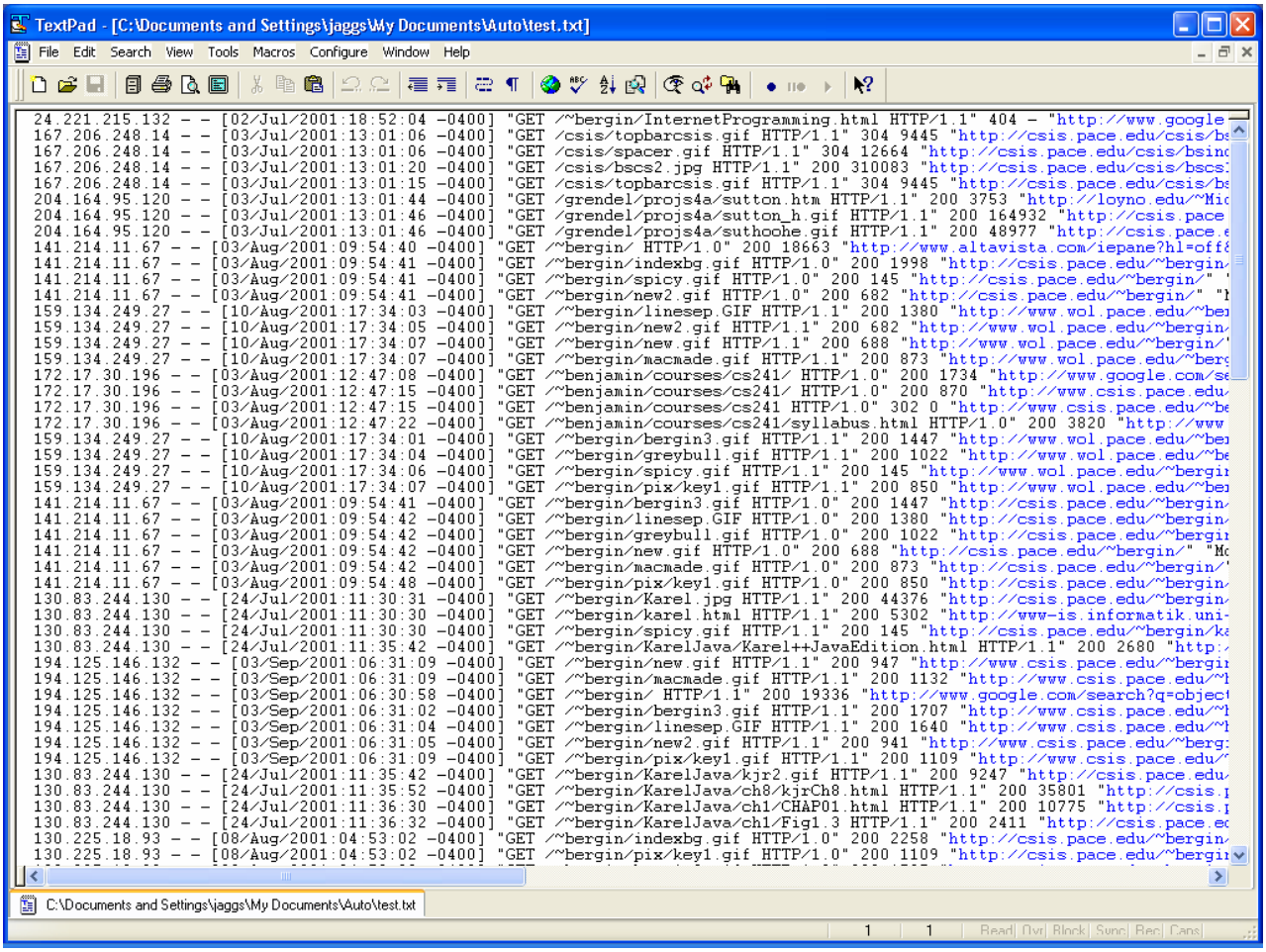
As the page loads links based on the http referrer information (keyword) are displayed in the top of the page.

Knowledge About New Users (KANU):

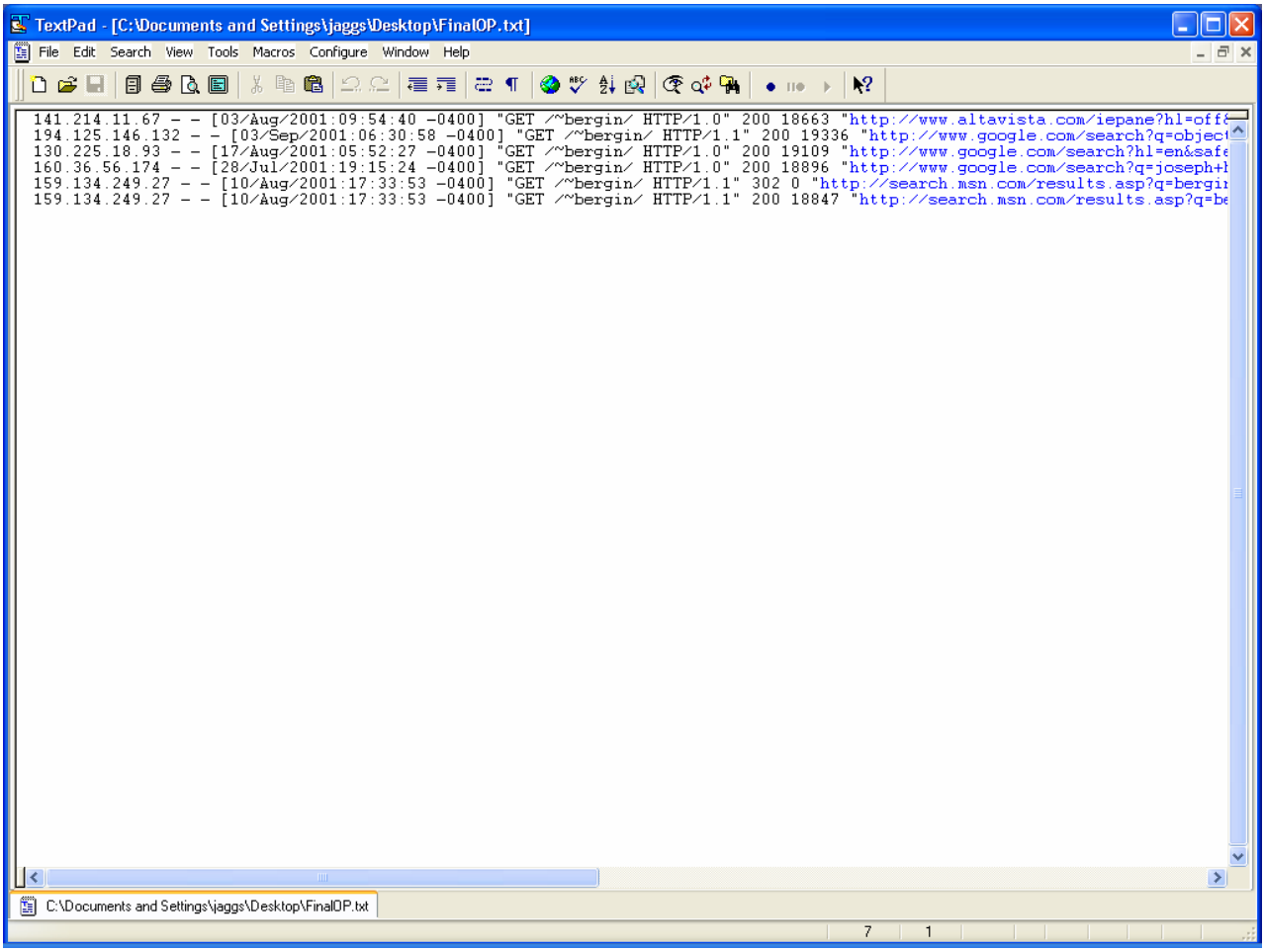
The web site administrator with his KANU tool can check if the user has been a serious surfer or not (time spent on the web site) and can keep track of this *new keyword* pertaining to the users'.

Hence, the administrator tool gives him/her an idea to understand any **new** activity on the site, which might need action. Any new keyword(s) with respect to information added or deleted on the site can be known by looking at the condensed log file. The tool will write the output to a new file indicating the activity. As a result a busy administrator need not take a look at his/her long log files, but just run the tool which will automatically analyzes the log files and reports the information about users of the website who have been serious surfers (based on the time spent by the user on the website, between 5 seconds to 30 minutes) on the website.

For Example: When the website has millions of hits, then it's a tedious process to analyze them



The condensed log file gives an idea of new users on the website and their keywords can be noted by the site administrator. The site administrator can use this new information pertaining to the users to improve the site.



Issues and Scope:

Cleaning the 412MB server log files has been a slow process. This can be tackled if the data on a faster system with more memory, hard drive space and processor speed. Using of grep on each and every case on the 412MB file has been time consuming. The application is run on Tomcat/Apache. It's a java based ProxyServelet. The faster is the host machine with more speed and memory the quicker is the experience.

Patterns of user behavior can be extracted when we have access to various server log files. A relation to describe the user, his activity, intentions can be established to identify valuable information. These no obvious relations can be used to predict, test, and conclude the information derived.

For example: The interest of a user can be understood if we know if the user was on both the CSIS and Lubin School of Business website. As a result, this kind of information can be helpful in understanding the users and improve the websites based on the new information. The vast data should be collected, understood and used to get the specific behavioral patterns which will be useful in providing better services. Today these kinds of patterns can be extracted in real world with real data like phone numbers, addresses, to predict if a person is a threat to the system or not.

Conclusion:

In this thesis it has been proved that machine learning can be used to understand the human behavior on the websites. As much as machine learning has been used to create automated industrial systems, this also proves that we can use AI successfully on server log files to extract patterns and create applications improving various performance parameters. The goal to understand server log files, user behavior on the website, keywords used, and their relevance to the site have been effectively addressed. The scope for further research is immense in data mining and AI. As a result new insights can be gained into specific user needs for information based on various parameters. AI is the holy grail of computer science. The ROI will be justified if the server log files are understood better to understand human beings and their online behavior.

The ability to understand the minds of people has been a challenge for all of us. This has been proved in our day to day life, global situations and terrorist attacks. Any rationale behind an action is justifiable by only the subject who has the prior knowledge of his/her actions. Understanding the patterns, signatures left behind, movements, can help us in creating solutions to new world problems, addressing the current problems and preventing future issues. Technology is our best friend, but the success of technology vastly depends on how close the system will think like humans. In many cases the learning agents are doing better job than human beings like “fly by wire” for Airbus / Boeing jets, BP GASOIL system.

However the case of understanding online behavior is more important today because we are seeking value for the investments made by us and expecting returns. Any small step towards such

an understanding will lead to some results which will really help in optimizing the performances of the websites, systems, human beings etc.

Bibliography

a. Artificial Intelligence- A Modern Approach

b. Java Servelt Programming

c. e-Profit

d. See5.0/C5.0 (www.rulequest.com)

e. Data Warehousing, Data Mining, & OLAP

f. World Wide Web