A Major Project Report On

EVALUATION OF WEBSITES USING MACHINE LEARNING ALGORITHMS

Submitted in partial fulfilment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

IN

SOFTWARE ENGINEERING

By

Prerana Yadav

(Roll No. 11/SWE/2k10)

Under the Guidance of

Dr. Ruchika Malhotra

Department of Software Engineering

Delhi Technological University, Delhi

Department of Computer Engineering

Delhi Technological University, Delhi

2010-2012

DELHI TECHNOLOGICAL UNIVERSITY

CERTIFICATE

This is to certify that the project report entitled EVALUATION OF WEBSITES USING MACHINE LEARNING ALGORITHMS is a bona fide record of work carried out by Prerana Yadav (11/SWE/2k10) under my guidance and supervision, during the academic session 2010-2012 in partial fulfilment of the requirement for the degree of Master of Technology in Software Engineering from Delhi Technological University, Delhi.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Dr. Ruchika Malhotra

Asst. Professor

Department of Software Engineering

Delhi Technological University

Delhi

JUNE-2012

DELHI TECHNOLOGICAL UNIVERSITY

ACKNOWLEDGEMENT

It's true and proved that behind every success, there is certainly an unseen power of Almighty God. He is the grand operator of all projects.

First of all, I thank my parents for their encouragement and persistent support which has helped me to do better in all of my endeavours.

I would like to express my deepest sense of gratitude and indebtedness to my guide and motivator **Dr. Ruchika Malhotra**, Assistant Professor, Department of Software Engineering, Delhi Technological University for her valuable guidance and support in all the phases from conceptualization to final completion of the project.

I wish to convey my sincere gratitude to **Prof. Daya Gupta**, Head of Department, and all the faculties of Computer Engineering Department, Delhi Technological University who have enlightened me during my project.

I humbly extend my grateful appreciation to my friends, in particular Ms. Poonam Goel, Mr. Nakul Pritam, Ms. Reena Negi & Mr. Prateek Negi whose moral support made this project possible.

Last but not the least; I would like to thank all the people directly and indirectly involved in successful completion of this project.

PRERANA YADAV

Roll No. 11/SWE/2k10

TABLE OF CONTENTS

Certificate .... **i**

Acknowledgement .... **ii**

Table of Contents .... **iii-iv**

List of Figures .... **v-vi**

List of Tables .... **vii**

Abstract .... **viii**

Chapter 1: Introduction **1-4**

1.1 Introduction 1

1.2 Motivation of the Work 2

1.3 Aim of the Work 3

1.4 Organization of Thesis 3

Chapter 2: Related Work **5-7**

Chapter 3: Research Background **8-18**

3.1 Web Page Measures 8

3.1.1 Page Composition Metrics 9

3.1.2 Page Formatting Metrics 10

3.1.3 Overall Page Quality or Assessment Metrics 11

3.2 Independent and Dependent Variables 11

3.3 Empirical Data Collection 14

3.3.1 Categorization of Websites as Good and Bad 15

Chapter 4: Research Methodology **19-42**

4.1 Methodology 19

4.2 Description of Tool 20

4.2.1 Module I –Webpage Processor 21

4.2.2 Module II – Metric Calculator 22

4.3 Machine Learning Algorithms for Data Analysis 32

4.3.1 Multilayer Perceptron 32

4.3.1.1 The Algorithm 32

4.3.1.2 Weight Training Calculation in Backward Phase 33
4.3.2 Naïve Bayes Classifier 34
4.3.3 Decision Tree 35
4.3.3.1 The Algorithm 35
4.3.4 Bagging 36
4.3.4.1 Aggregation 37
4.3.4.2 Bootstrapping 37
4.3.5 Random Forest 38
4.3.5.1 The Algorithm 39
4.3.5.2 Out-Of-Bag Error Estimation 40
4.3.6 AdaBoost 41
4.3.6.1 The Algorithm 41
4.3.7 Random Tree 42
4.3.8 Decision Table 42
Chapter 5: Results Analysis 43-73
5.1 Descriptive Analysis 45
5.2 Logistic Regression Analysis 47
5.3 Multilayer Perceptron Analysis 47
5.4 Naïve Bayes Analysis 50
5.5 Decision Tree Analysis 53
5.6 Bagging Analysis 56
5.7 Random Forest Analysis 59
5.8 AdaBoost Analysis 62
5.9 Random Tree Analysis 64
5.10 Decision Table Analysis 67
5.11 Model Evaluation 69
Chapter 6: Conclusions and Future Work 74-75
6.1 Application of Work 75
6.2 Future Work 75
References 76-79

LIST OF FIGURES

Figure 3.1: Classification of websites based on Pixel Awards 2009 Data 15
Figure 3.2: Classification of websites based on Pixel Awards 2010 Data 16
Figure 3.3: Classification of websites based on Pixel Awards 2011 Data 16
Figure 4.1: Flowchart of Methodology 20
Figure 4.2: The interface for Web Metrics Analyzer 21
Figure 4.3: Table Count metric estimation for dataset of 2009 23
Figure 4.4: Link Count metric estimation for dataset of 2009.. 24
Figure 4.5: Graphic Count metric estimation for dataset of 2009 25
Figure 4.6: List Count metric estimation for dataset of 2009 26
Figure 4.7: Page Title Length metric estimation for dataset of 2009.................................... 27
Figure 4.8: Division Count metric estimation for dataset of 2009... 28
Figure 4.9: Word Count metric estimation for dataset of 2009 29
Figure 4.10: Body Text Words metric estimation for dataset of 2009................................... 30
Figure 4.11: Page Size metric estimation for dataset of 2009 31
Figure 4.12: Architectural Graph of Multilayer Perceptron with two hidden layer 32
Figure 4.13: A Forest of Trees 39
Figure 5.1: ROC Curve for Model I using Multilayer Perceptron... 49
Figure 5.2: ROC Curve for Model II using Multilayer Perceptron.. 49
Figure 5.3: ROC Curve for Model III using Multilayer Perceptron 50
Figure 5.4: ROC Curve for Model I using Naïve Bayes Classifier 52
Figure 5.5: ROC Curve for Model II using Naïve Bayes Classifier 52
Figure 5.6: ROC Curve for Model III using Naïve Bayes Classifier 53
Figure 5.7: ROC Curve for Model I using Decision Tree 55
Figure 5.8: ROC Curve for Model II using Decision Tree 55
Figure 5.9: ROC Curve for Model III using Decision Tree 56
Figure 5.10: ROC Curve for Model I using Bagging... 57
Figure 5.11: ROC Curve for Model II using Bagging 58
Figure 5.12: ROC Curve for Model III using Bagging 58
Figure 5.13: ROC Curve for Model I using Random Forest 60

Figure 5.14: ROC Curve for Model II using Random Forest 61
Figure 5.15: ROC Curve for Model III using Random Forest 61
Figure 5.16: ROC Curve for Model I using AdaBoost 63
Figure 5.17: ROC Curve for Model II using AdaBoost 63
Figure 5.18: ROC Curve for Model III using AdaBoost 64
Figure 5.19: ROC Curve for Model I using Random Tree 65
Figure 5.20: ROC Curve for Model II using Random Tree 66
Figure 5.21: ROC Curve for Model III using Random Tree 66
Figure 5.22: ROC Curve for Model I using Decision Table 68
Figure 5.23: ROC Curve for Model II using Decision Table 68
Figure 5.24: ROC Curve for Model III using Decision Table 69

LIST OF TABLES

Table 3.1: Page Composition Metrics 9
Table 3.2: Page Formatting Metrics 10
Table 3.3: Overall Page Quality or Assessment Metrics 11
Table 3.4: Metrics Selected For Study 12
Table 3.5: Categorization of websites 17
Table 3.6: Number of web pages in all the datasets for all categories 18
Table 4.1: Structure of Decision Table 42
Table 5.1: Descriptive Statistics of dataset 2009 45
Table 5.2: Descriptive Statistics of dataset 2010 46
Table 5.3: Descriptive Statistics of dataset 2011 46
Table 5.4: Website Prediction of Logistic Regression for Model I, II and III 47
Table 5.5: 10-cross validation results for models using Logistic Regression 47
Table 5.6: Website Prediction of Multilayer Perceptron for Model I, II and III 48
Table 5.7: 10-cross validation results for models using Multilayer Perceptron 48
Table 5.8: Website Prediction of Naïve Bayes Classifier for Model I, II and III 51
Table 5.9: 10-cross validation results for models using Naïve Bayes Classifier 51
Table 5.10: Website Prediction of Decision Tree for Model I, II and III 54
Table 5.11: 10-cross validation results for models using Decision Tree 54
Table 5.12: Website Prediction of Bagging for Model I, II and III 57
Table 5.13: 10-cross validation results for models using Bagging 57
Table 5.14: Website Prediction of Random Forest for Model I, II and III 59
Table 5.15: 10-cross validation results for models using Random Forest 60
Table 5.16: Website Prediction of AdaBoost for Model I, II and III 62
Table 5.17: 10-cross validation results for models using AdaBoost 62
Table 5.18: Website Prediction of Random Tree for Model I, II and III 65
Table 5.19: 10-cross validation results for models using Random Tree 65
Table 5.20: Website Prediction of Decision Tablet for Model I, II and III 67
Table 5.21: 10-cross validation results for models using Decision Table 67
Table 5.22: Summarized results of 10-cross validation of all the models 71

ABSTRACT

As World Wide Web is getting more prevalent day by day, the need for quality websites is becoming necessary. Evaluating website quality is an essential, but a debatable subject, where there are few ways to analyze and evaluate the quality of the website in quantitative form. Various guidelines have been proposed, however it is not clear how to implement them. Since metrics are crucial source of information in decision making, web metrics are used to estimate the quality of the web engineered product or the process to build it. It is requisite to continuously assess and evaluate the websites and subsequently to make improvements over those evaluations in order to enhance the website quality.

In this research, we have computed nine quantitative web measures for each website using an automated Web Metrics Analyzer tool developed in JAVA programming language and derived a relationship between these metrics and website quality. We have examined a collection of 2678 web pages from 255 expert reviewed websites from different categories of Pixel Awards 2009 to 2011 for the assessment of the quality of websites into good or bad. In order to analyse the results, we have used logistic regression and machine learning techniques like Multilayer Perceptron, Naïve Bayes, Decision Tree, Bagging, Random Forest, AdaBoost, Random Tree & Decision table on the dataset. The results show that Naïve Bayes method has the highest area under curve (computed using Receiver Operating Curve analysis) within 0.843-0.923 in all the three datasets. Thus the performance of Naïve Bayes method model is better than all other compared models.

Dedication

I dedicate this thesis

to my family, my teachers and my friends for

supporting me all the way & doing all thee

wonderful things for me.

Chapter 1

INTRODUCTION

1.1 Introduction

There is an enormous amount of information available on the Internet which is growing at a tremendous rate. With such increasing use of World Wide Web, the need for quality websites is necessary. As the web pages can be easily created without any training, much of the information on the Web is of poor quality, which makes it impossible for an individual to find the right or relevant information. Thus, it is requisite to continuously assess and evaluate the websites and subsequently to make improvements over those evaluations in order to enhance the website quality.

There has been a lot of serious discussion about what makes a good website and despite the ample guidelines for building a usable website, developing quality websites is still a problem [10]. Websites have a property of constantly changing themselves in the way information is provided to the users. The quality of a website is represented in terms of the amount of information presented in one page, the relevancy of information provided, the user-friendliness of website and many other properties. As metrics are crucial source of information in decision making, a number of various metrics of web page composition and layout have been proposed that contribute to the usability and goodness of website. Website attributes can be also classified into two major types: internal and external attributes. Internal attributes are those characteristics that can be directly measured while external attributes are those we like to measure or know. Internal attributes can be used to help us know the external one. However, the relation between internal and external attributes is not simple and direct. For example, the number of links in a web page is a size internal metric that can be directly measured and it has relation with several external metrics such as size, complexity, quality, etc [1]. These attributes can be used to predict the usability of websites.

1.2 Motivation of the Work

The motivation for the work done in this research is the difficulty in the analysis of the websites faced by various developers. Many detailed website design guidelines have been developed for both general user interfaces and web page design [15]. These design guidelines are often stated at such a high level, that it is unclear how to implement them. As a result, designers have found number of difficulties following these guidelines where there is no opinion about the correctness of these guidelines [5] [12] [15].

Although most eminent web sites are created by professional design firms, there are many smaller sites that are built by people with little design experience or training. As a result, websites that belong to small businesses or organizations often have quality below a satisfactory level. These poorly designed websites can lead to lost productivity and revenue. The question of how to improve the design of websites is thus very important. In this work, we investigate the following issues:

- How web page metrics are related to quality of websites?
- How accurately and precisely do the web page measures predict the quality of websites?
- Is the performance of machine learning methods better than the logistic Regression technique?

1.3 Aim of the Work

Our research goal is to evaluate the quality of websites by employing the relationship between the web page metrics and goodness of website and to examine the differences between websites with quality and without quality. Software metrics has not been applied much in the field of World Wide Web [23]. We have used the webpage design metrics for evaluating website quality. In our last study, we evaluated a dataset of 87 web pages, considering only homepages and applied machine learning techniques on it.

In this study, we extend our previous work by examining a collection of 2678 web pages from 255 expert reviewed websites from different categories of Pixel Awards 2009 to 2011 for the assessment of the quality of websites into good or bad. The dataset of web pages is sufficiently large to prove that the web page metrics have the ability to predicting the quality of websites. For this study, we developed an automated Web Metrics Analyzer tool in JAVA programming language that is used to compute 9 quantitative measures having to do with page composition, layout, amount of information, and size (e.g., number of words, links, and lists) for each webpage. These metrics make a subset of a set of web page metrics which are related to the quality of web page design. We applied and compared logistic regression and various machine learning techniques over the metric estimations calculated for the collection of websites and predicted the model having the highest accuracy results. The results of this model allow us to predict precisely if a website is categorized as good or bad.

1.4 Organization of Thesis

The remainder part of thesis is organized in the following chapters:

Chapter 2: Related Work

This section highlights the introduction to the related work that has been done in the field of evaluation of website.

Chapter 3: Research Background

This section describes the research background in detail, i.e. it gives the brief introduction of the various quantitative web interface measures and the independent and dependent variables selected for our study.

Chapter 4: Research Methodology

The empirical datasets and their characteristics are discussed in this section with the brief introduction of the tool used for the metrics estimation process. It also presents the various machine learning algorithms to be used for data analysis.

Chapter 5: Results Analysis

We evaluate and judge the performance of our results. This section discusses the comparative analysis of results of applying various machine learning algorithms onto the collected dataset.

Chapter 6: Conclusions and Future Work

Conclusions are drawn here. This section also incorporates the scope of future integration.

References: This section gives the reference details including the list of materials (URLs, Books, and research papers) used and studied.

Chapter 2

RELATED WORK

A lot of existing work has been done on evaluating web page quality, but most quantitative methods for evaluating web sites focus on statistical analysis of usage patterns in server [4][6][8]. Traffic-based analysis (e.g., pages-per-visitor or visitors-per-page) and time-based analysis (e.g., click paths and page-view durations) provide data that must be interpreted in order to identify usability problems. The analysis based on such data is quite uncertain since web server logs provide incomplete traces of user behavior, and because timing estimates may be skewed by network latencies.

The above work focuses more on navigation history; explicitly clicked links and the time spend on a web site. Server logs are problematic because they only track unique navigational events (e.g., do not capture use of back button) and thus are hard to understand because of caching. Another method for evaluating web pages of user interest automatically investigates various factors in a user's browsing behavior such as number of scrolls, form input, search text etc [21].

Other approaches were inspection-based that rely on assessing static HTML according to a number of pre-determined guidelines, such as whether all graphics contain ALT attributes that can be read by screen readers [17]. For example, WebSAT (Web Static Analyzer Tool) is used to check the accessibility issues (i.e., support for users with disabilities), forms use, download speed, maintainability, navigation and readability of Web pages. There are many other techniques that compare quantitative web page attributes – such as the number of links or graphics – to thresholds [20]. However, there are no clear thresholds established for a wider class of quantitative Web page measures.

Simulation has also been used for web site quality evaluation. For example, a simulation approach has been developed for generating navigation paths for a site based on content similarity among pages, server log data, and linking structure [4]. The simulation models hypothetical users who are traversing the site from described start pages, making use of information "scent" (i.e., common keywords between the user's goal and linked pages content) to make decisions related to navigation. The approach does not consider the impact of various web page attributes, such as the amount of text or layout of links.

The most closely related work is done in Ivory et.al [10] [11] which provides preliminary analysis of collection of web pages and captures various web metrics associated with the rated websites, and predicts how the pair-wise correlations are manifested in the layout of the rated and unrated sites pages. This work does not apply various machine learning algorithms to predict the best suited model that can provide high accuracy.

The approach presented by G. Velayathan and S. Yamada [21] analyzes the user logs metrics such as number of scrolls, form input, search text etc. and extracts effective rules to evaluate web pages using a machine-learning method known as decision tree. A client side logging/analyzing tool GINIS is used to automatically evaluate web pages using these learned rules. Similarly, M. Zorman et.al [24] has proposed an algorithm to find the good or relevant websites for keywords provided by the user. They developed an intelligent search tool which employs TFIDF heuristics for finding term frequency and decision tree machine learning algorithm for automatically evaluation of the websites.

Another approach was based on applying Ranking SVM [13] [14] which is used to extract evaluation criteria from evaluation data for automated web site evaluation. It chooses the evaluation criteria which are the discriminant functions learned from a set of ranking information and evaluation features such as freshness, accuracy of spelling and grammar, top page's global link popularity collected automatically by web robots. However, it does not consider the other algorithms for the website evaluation.

The quality of a website can be defined in terms of functional as well as non-functional properties. K. M. Khan [25] has derived the non-functional attributes such as reliability, usability, efficiency, security and assessed them. The work done in [25] adopts a Goal-Question-Metric (GQM) approach to derive quality metrics. It defines the goals that are needed to be measured, then it develops the questions derived from goals that are required to determine if the goals are fulfilled, and finally, their measurements are the answers of the questions which are known as metrics. For instance, questions related to the goal failure rate could be: what is the percentage of incorrect links on the page?

Chapter 3

RESEARCH BACKGROUND

Our study aims at determining the effect of various webpage attributes on the goodness of the web pages. The foremost task is to select the web page metrics which are to be considered as independent variables and the data analysis techniques to analyze the dataset.

In this chapter, section 3.1 introduces the various web interface measures proposed by Ivory et.al [10] based on the guidelines stated in various literature and section 3.2 defines the independent and dependent variables used for our study.

3.1 Web Page Measures

Software metrics can be applied to all the phases of software development life cycle from beginning, when cost must be estimated, to monitoring the reliability of the products and sub-products and end of the product, even after the product is deployed and is in operational environment [18].

Study of websites is comparatively a new convention to quality management which makes the task of measuring web sites quality very important. Since metrics are crucial source of information in the field of decision making, a large number of web metrics have been proposed in the last decade to evaluate the structural quality of a web page quantitatively.

Based on the various guidelines mentioned in [12] [15] and survey of Web design literature, Ivory et.al developed various page level and site level measures to assess 62 features associated with effective design and goodness of webpage such as amount of text on a page, fonts, etc. The classification of those web page measures is given below:-

3.1.1 Page composition Metrics

This type of metrics assesses the attributes that correspond to the composition of page.

Table 3.1: Page Composition Metrics

Metric	Description
Number of Words	Total words on a page
Body Text Words	Words that are body vs. display text (i.e., headers)
Link Text Words	Total words in links
Number of Links	Links on a page
Length of Link Text	Words in the text for a link
Redundant Links	Repeated links on a page
Embedded Links	Links embedded in text on a page
Wrapped Links	Links spanning multiple lines
Within-page Links	Links to other areas of the same page
Readability	Reading level of text on a page
Number of !'s	Exclamation points on a page
Content Percentage	Portion of page devoted to content
Navigation Percentage	Portion of page devoted to navigation
Page Title Length	Words in the page's title
Number of Graphics	Total images on a page
Page Size	Total bytes for the page and images

Image Size	Number of pixels in an image
Total Graphics Size	Total bytes for images
Animated Elements	Animated images and scrolling text

3.1.2 Page Formatting Metrics

These metrics assess the layout of elements on the page.

Table 3.2: Page Formatting Metrics

Metric	Description
Font Styles	Types of fonts (e.g., serif vs. sans serif) employed
Font Point Size	Font sizes employed (e.g., 9pt vs. 14pt)
Text Emphasis	Total emphasized (e.g., bold, italicized, and capitalized) text
Emphasized Body Text	Total emphasized (e.g., bold, italicized, and capitalized) body text
Number of Font Faces	Total font faces employed
Number of Font Sizes	Total font sizes employed
Screen Coverage	Total screen area covered (i.e., non whitespace)
Number of Screens	Number of vertical and horizontal scrolls required
Text Clustering	Text areas highlighted with color or bordered regions
Text in Clusters	Words in text clusters
Text Positions	Changes in text positions from flush left
Number of Lists	Lists on a page
Number of Rules	Vertical and horizontal rules on a page
Number of Colors	Total colors employed

Line Length	Width of text lines on a page
Leading	Spacing between consecutive text lines on a page
Frames	Use of frames

3.1.3 Overall Page Quality or Assessment Metrics

These metrics are the high level characteristics that correspond to the overall quality of the page.

Table 3.3: Overall Page Quality or Assessment Metrics

Metric	Description
Information Quality	Content appropriateness (i.e., relevance, language, and tone)
Image Quality	Image appropriateness and optimization (size and resolution)
Link Quality	Link clarity (scent) and relevance
Layout Quality	Aethestics, alignment and balance
Download Speed	Time for a page to fully load

3.2 Independent and Dependent Variables

The dataset comprises of 10 measures to be used for web pages, one dependent and nine independent variables. These variables cover those attributes that can be computed automatically. Out of all above mentioned metrics, Table 3.4 describes the 9 metrics that we have selected as variables for our study. We developed a Web Metric Analyzer tool developed in JAVA technology to compute these metrics which has been explained in later chapter. There are various techniques that are used for incorporating the correlation of independent variables. We have used Feature sub selection technique for reducing data dimensionality provided in WEKA tool [22]. Correlation based Feature Selection (CFS) technique is applied to select the best predictors out of all the independent variables in the dataset [31]. The best combinations of independent variable are searched through all possible combinations of variables. CFS provides with the good feature sets that are highly correlated with the dataset [32]. It evaluates the best of a subset of variables by considering the individual predictive ability of each feature along with the degree of redundancy between them.

Table 3.4: Metrics Selected For Study

Metrics	Description
Word Count	Total Words on a page
Body Text Words	Words that are body vs. display text
Page Size	Total bytes of the page and images
Table Count	Number of tables present on the webpage
Graphics Count	Total images on a page
Division Count	Divisions on a page
List Count	Lists on a page
Number of Links	Links on a page
Page Title length	Words in Page Title

The description of the attributes calculated by the tool is:

a) Word Count

Total number of words on a page is taken. This attribute is calculated by counting total number of words on the page. Special characters such as & / are also considered as words.

b) Body text words

This metric counts the number of words in the body Vs display text. In this, we calculate the words that are part of body and the words that are part of display text separately. The words can be calculated by simply counting the number of words falling in body.

c) Page size

It refers to the total size of the web page and can be found in properties option of the web page.

d) Table Count

This metric gives the number of tables used in making a web page.

e) Graphic Count

This refers to the total number of images on a page. And can be calculated by counting the total number of images present on the page. It has been analyzed that usable and good quality pages contain more images which contribute to the larger page size.

f) Division Count

This metric can be calculated by analyzing the number of divisions in a webpage.

g) List Count

This metric can be calculated by counting total number of ordered and unordered list present on a web page.

h) Link Count

These are the total number of links on a web page and can be calculated by counting the number of links present on the web page.

i) Page title length

This refers to the words in the page title and can be calculated by counting the total no of words in the page title.

The dependent variable is website categorization which takes two values: good or bad based on the judges rating of Pixel Awards for a specific website.

3.3 Empirical Data Collection

We investigated the web-pages collected from Pixel Awards website. The Pixel Awards are the website award, annually honoring compelling sites that have shown excellence in web design and development which was established by Erick & Lisa Laubach in 2006.

The Pixel Awards judges are proven innovators in their respective fields with broad web expertise and a knack for spotting extraordinary talent with fairness and accuracy as described in Pixel Awards [16]. The websites placed in 24 categories are judged on the basis of creative and technical blend of impeccable graphic design, artistry, technological expertise, and a powerful, stimulating user experience [16]. These sites are the best of the web, thus each site for its respective category is evaluated for innovation, content, navigation, visual design, functionality and overall site experience.

For our study, we collected 3 sets of data from 20 categories of Pixel Awards for each year from 2009 to 2011. The different categories are Agency, Apps, Animation, Art, Blogs, Commerce, Community, Experimental, Fashion, Funny, Food & Beverage, Geek, Green, Movies, Non-Profit, Personal, Sports, Student, Travel, Weird. We used A1 Website Download which is configured to crawl 0-level i.e. homepages and 1-level pages from each site; thus, we collected homepages and 1-level web pages for each website. In this way, we have the dataset for 2678 web pages from 255 websites. There are 945 web pages in 79 websites nominated in year 2009, 920 web pages in 90 websites from year 2010, and 813 web pages in 86 websites nominated in year 2011.

3.3.1 Categorization of Websites as Good and Bad

There are 2 awards given in each category, one is chosen by judges as winner, and another is People's Champ Winner. We have considered the winner websites in all the categories as good and all the other nominee websites as bad. Thus, we have 19 good classified and 60 bad classified web pages in dataset of year 2009 , 31 good and 59 bad classified websites in year 2010, and 33 good and 53 bad classified websites in year 2011 as shown in figure 3.1, 3.2 and 3.3 respectively.

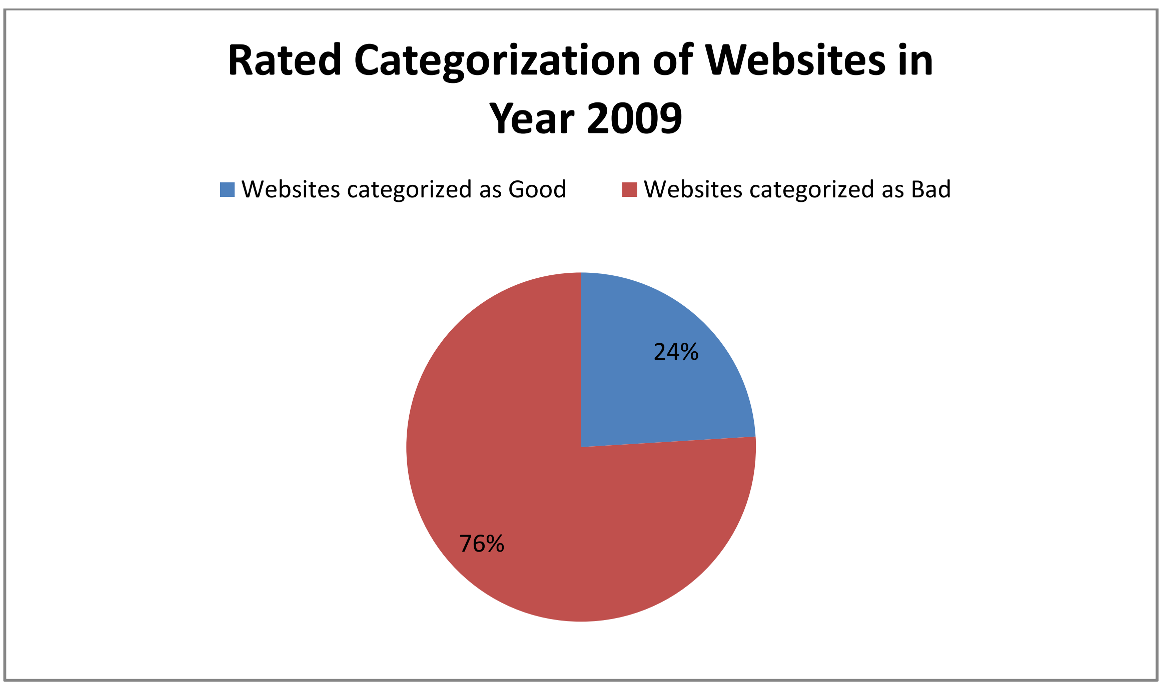

Figure 3.1: Classification of websites based on Pixel Awards 2009 Data

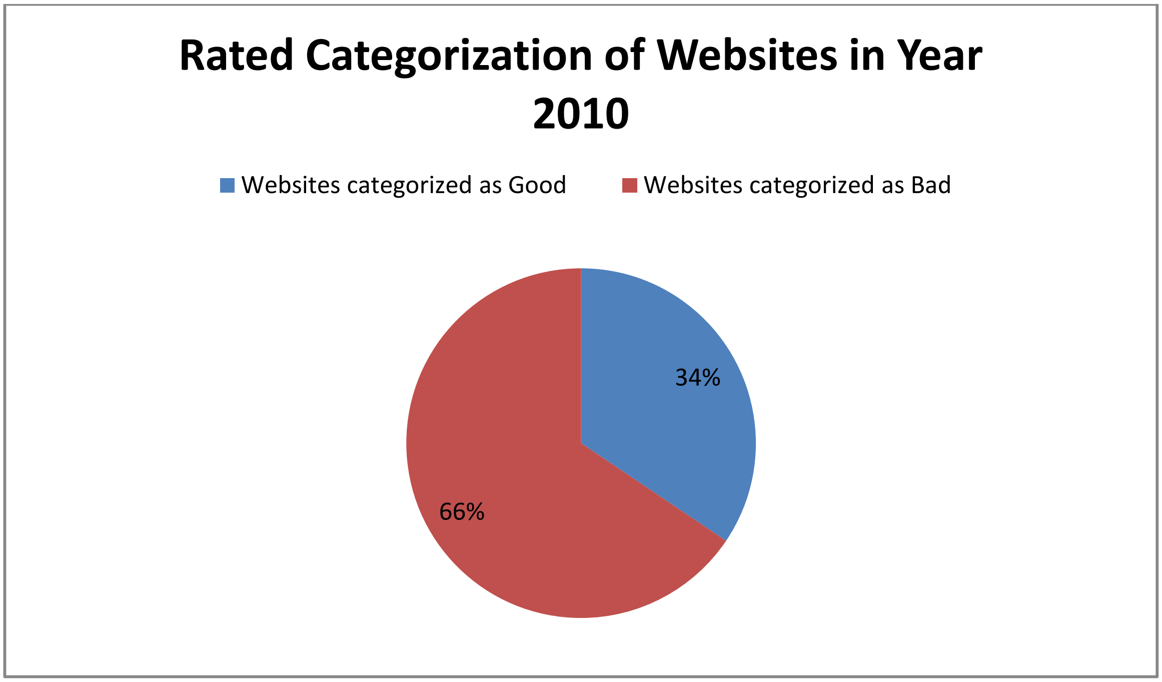

Figure 3.2: Classification of websites based on Pixel Awards 2010 Data

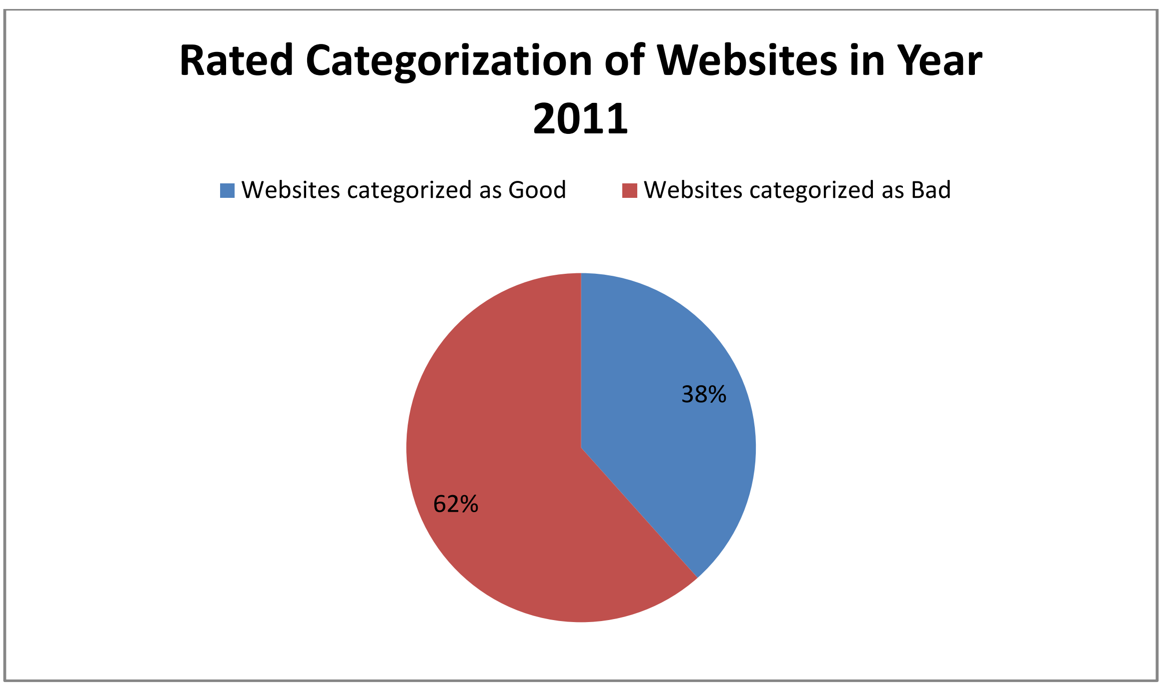

Figure 3.3: Classification of websites based on Pixel Awards 2011 Data

Table 3.5 shows the number of good and bad classified websites in 3 year data of Pixel Awards 2009-2011.

Table 3.5: Categorization of websites

	Websites 2009	Websites 2010	Websites 2011
Good	19	31	33
Bad	60	59	53

Table 3.6 shows the number of web pages for each category in all the three datasets. These web pages belonging to different websites are stored as html source files in the system, which are used to calculate the 9 web page metrics considered for estimation over these source files using the tool described in next chapter.

Table 3.6: Number of web pages in all the datasets for all categories

Category Name	**Number of Web pages**		
	Dataset 2009	**Dataset 2010**	**Dataset 2011**
Agency	60	35	73
Animation	43	29	37
Apps	0	5	8
Arts	49	88	51
Blogs	0	38	103
Commerce	42	22	57
Community	0	82	45
Experimental	33	6	29
Fashion	47	101	61
Food & Beverage	0	79	40
Funny	56	0	19
Games	14	22	25
Geeks	31	30	14
Green	19	90	24
Magazines	82	49	48
Movies	45	0	0
Non-Profit	71	70	64
Personal	85	38	18
Sports	63	15	12
Student	64	52	47
Travel	86	48	38
TV	39	0	0
Weird	16	21	0
Total Webpages	**945**	**920**	**813**

Chapter 4

RESEARCH METHODOLOGY

4.1 Methodology

Our methodology employs the quantitative web-page attributes (number of links, words etc.) to compare the goodness of the web-pages and to construct a model for predicting the class of website as good or bad. Figure 4.1 shows the flowchart of methodology.

The flowchart of methodology shows the 3 sections. The first section is Empirical Data Collection which involves 3 steps. The first step is to select the websites nominated in different categories in 3 years (2009-2011) from Pixel Awards for which metrics estimations are to be calculated. The next step is to use a crawler to download and store the web pages belonging to all these websites. Then, we use the web metric analyzer to calculate the different metrics for these websites.

The second section is web metrics analyzer which includes 2 modules: web page processor to preprocess the source files to make them suitable for metric calculation and metric calculator which calculate the metrics estimations.

The third section is result analysis which includes the analysis and comparison of data using machine learning algorithms explained in further section. The comparison shows which algorithm gives better results compared to others. This predicts the model that can be used for further evaluations of the websites.

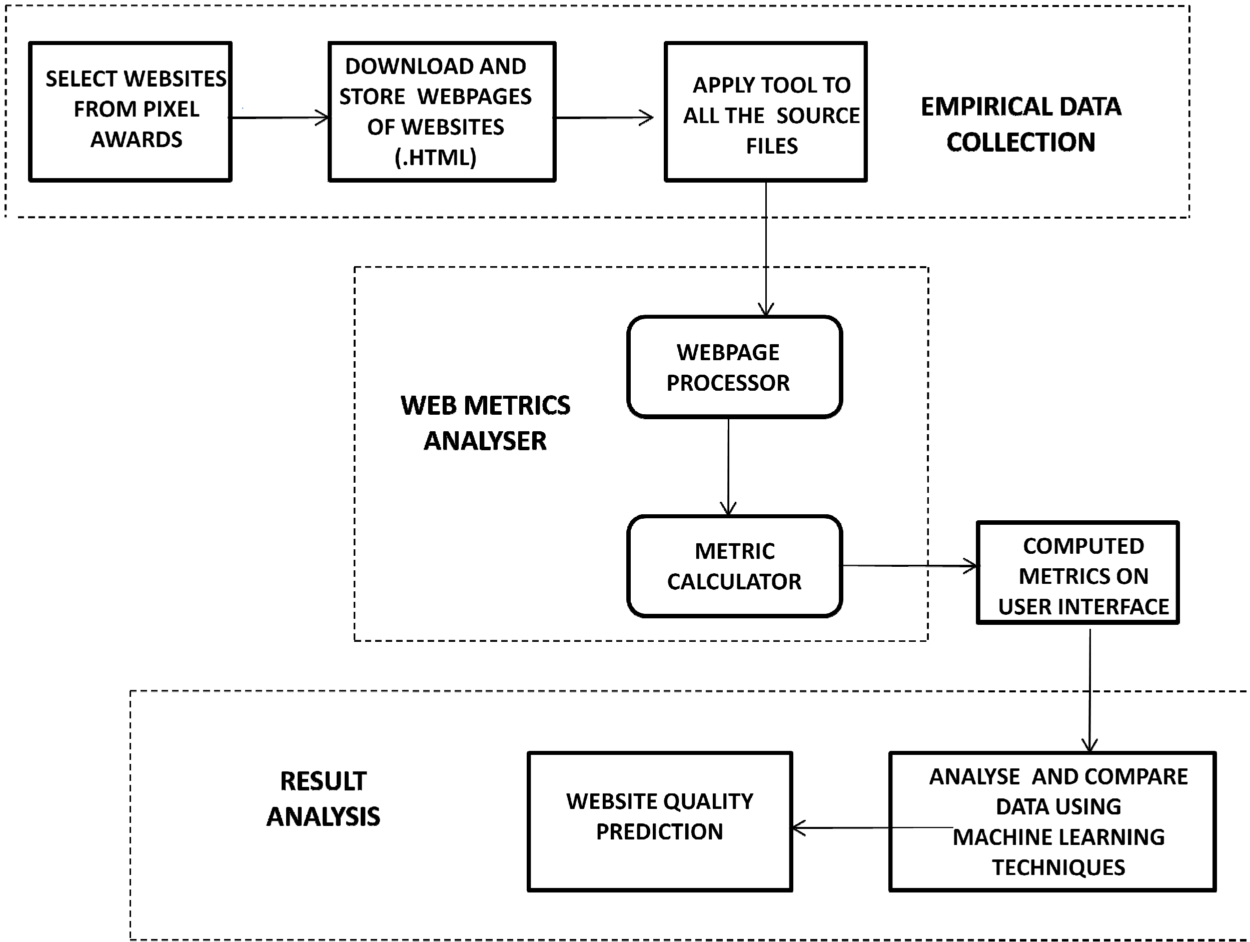

Figure 4.1: Flowchart of Methodology

4.2 Description of Tool

We have developed Web Metrics Analyzer, a JAVA based automated metrics analyzing tool that calculates 9 metrics to determine which of these attributes are correlated with the goodness and usability of the web page. The idea is to automatically collect information about the web pages that gives an idea of the flavor of the document. It demonstrates a way of rating web pages automatically for information content.

The tool works by taking an array of html source files for which metrics are need to be computed. The interface of the tool is shown in figure 4.2.

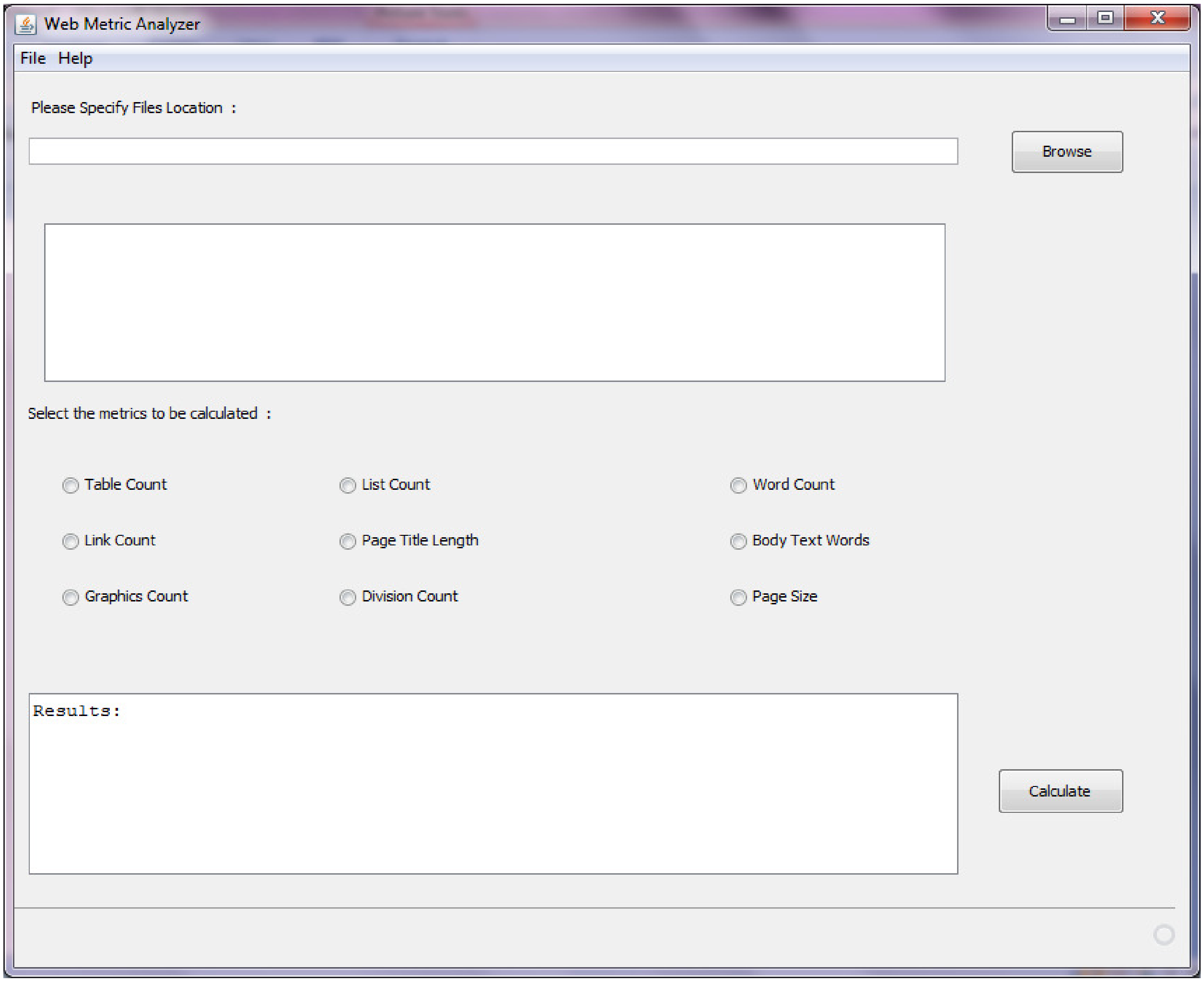

Figure 4.2: The interface for Web Metrics Analyzer

There are two main modules present in the tool that helps estimating the metric values for the html source files – Webpage Processor and Metric Calculator. The description of both the modules is given below:

4.2.1 Module I - Webpage Processor

The Webpage Processor module pre-processes the source files to make it fit for the Metrics Calculator module to work upon. This module conducts the following pre-processing on the source:

1. It tokenizes the source files and organizes separate web pages as separate arrays.
2. It also organizes different symbols like < > , ! # . as separate tokens as those are very important in measuring metrics.

The input to this module is a file name array that corresponds to the html source files that this module needs to process.

The output of this module is:

1. A 2-Dimensional array which contains tokens for the web pages (html files), to be used for metrics calculation.
2. A source file name array whose elements correspond to each row of tokens generated for the web pages.

4.2.2 Module II - Metric Calculator

The Metric Calculator module takes use of the 2-Dimensional array of tokens for web pages to estimate the particular metric. We need to select 1 metrics out of all the metrics to be computed.

The input to this module is:

1. A 2-Dimensional array of tokens obtained from the Module I.
2. A source file name array.
3. Selection of the metric to be calculated out of the 9 metrics.

The output of this module is an estimation of selected metric for all the source files. This estimation can be saved for the further reference. The graphical representation of metrics estimation using the web metric analyzer for all the websites of year 2009 is shown in figure 4.3 – 4.11.

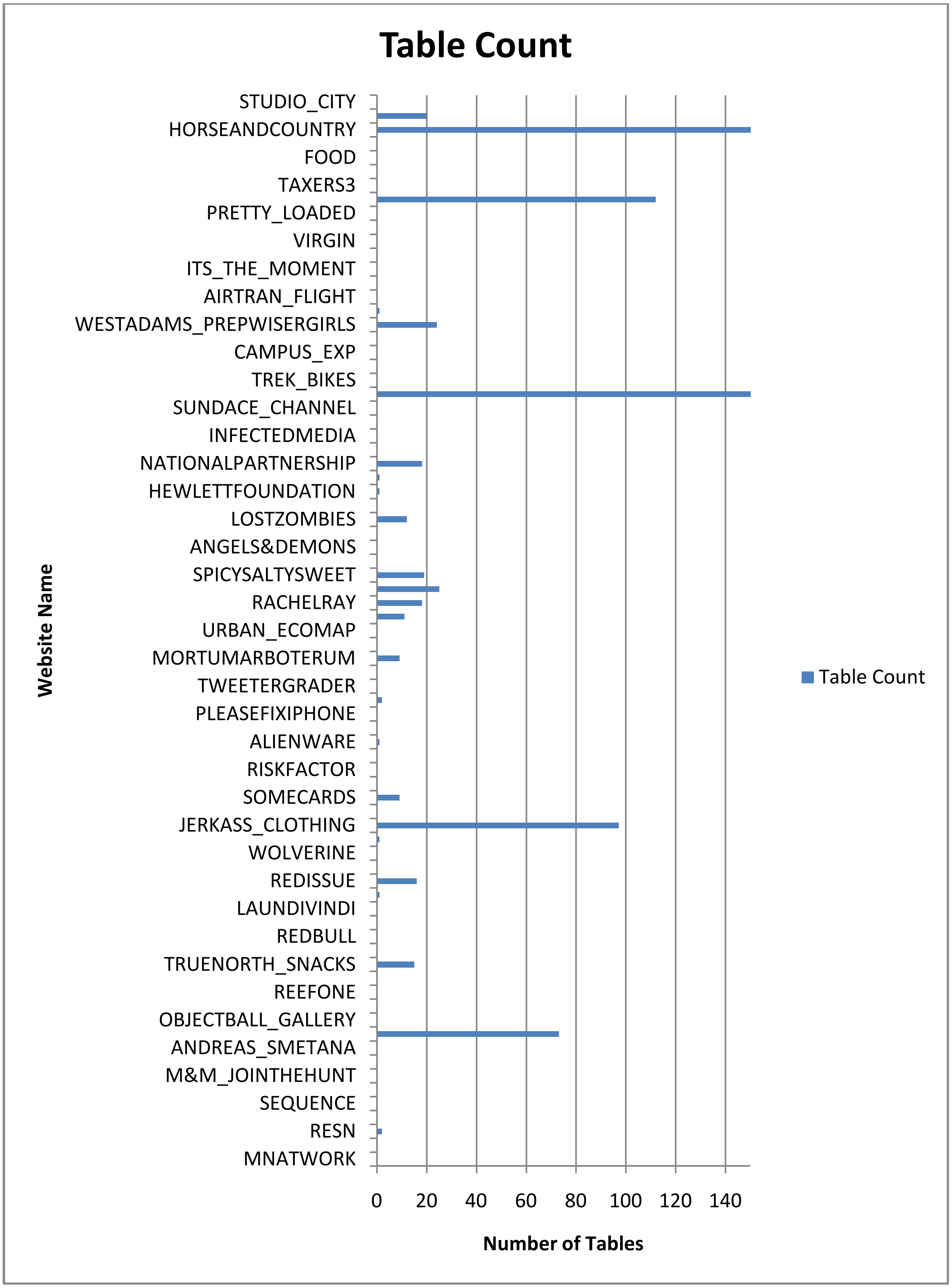

Figure 4.3: Table Count metric estimation for dataset of 2009

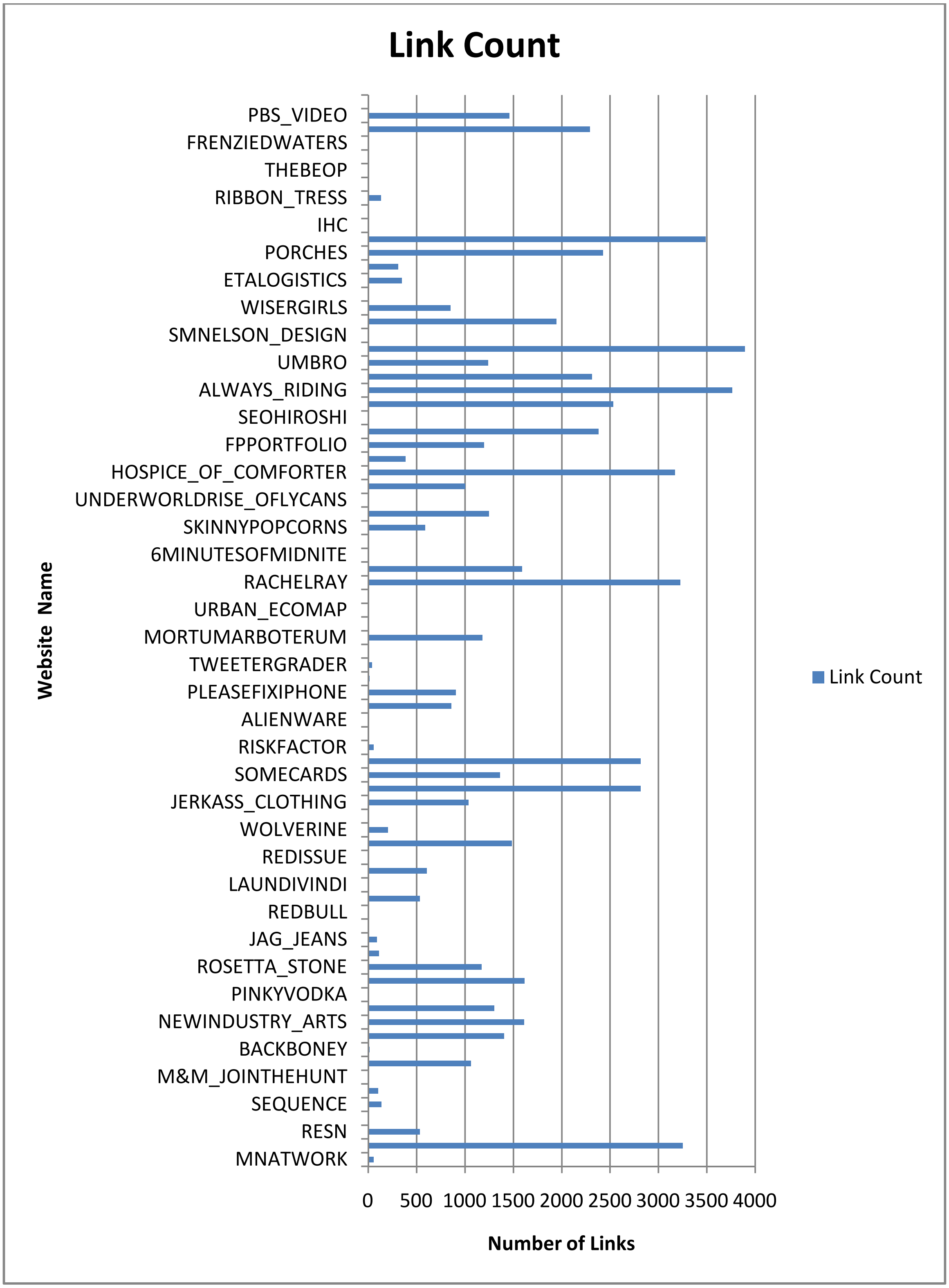

Figure 4.4: Link Count metric estimation for dataset of 2009

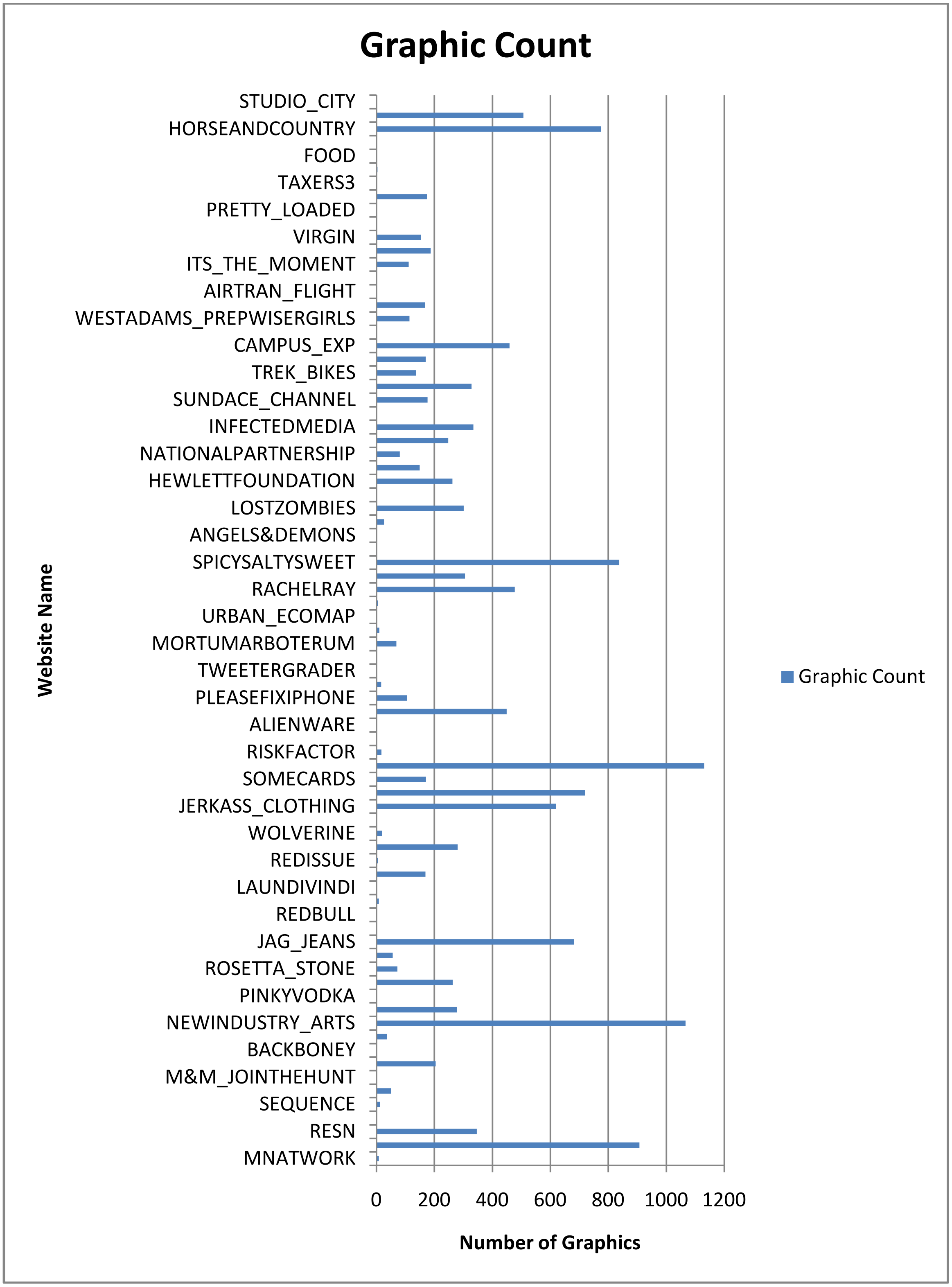

Figure 4.5: Graphic Count metric estimation for dataset of 2009

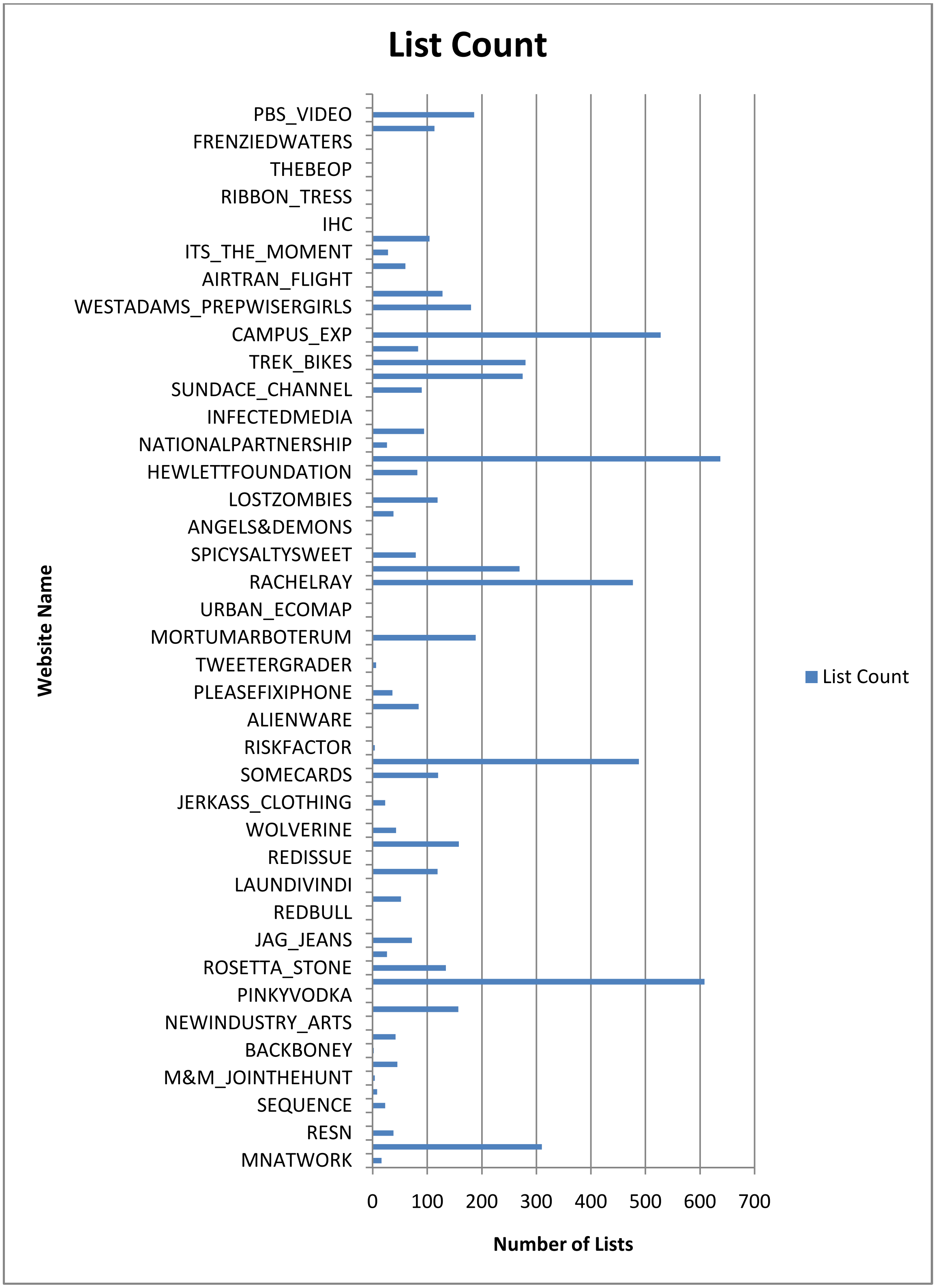

Figure 4.6: List Count metric estimation for dataset of 2009

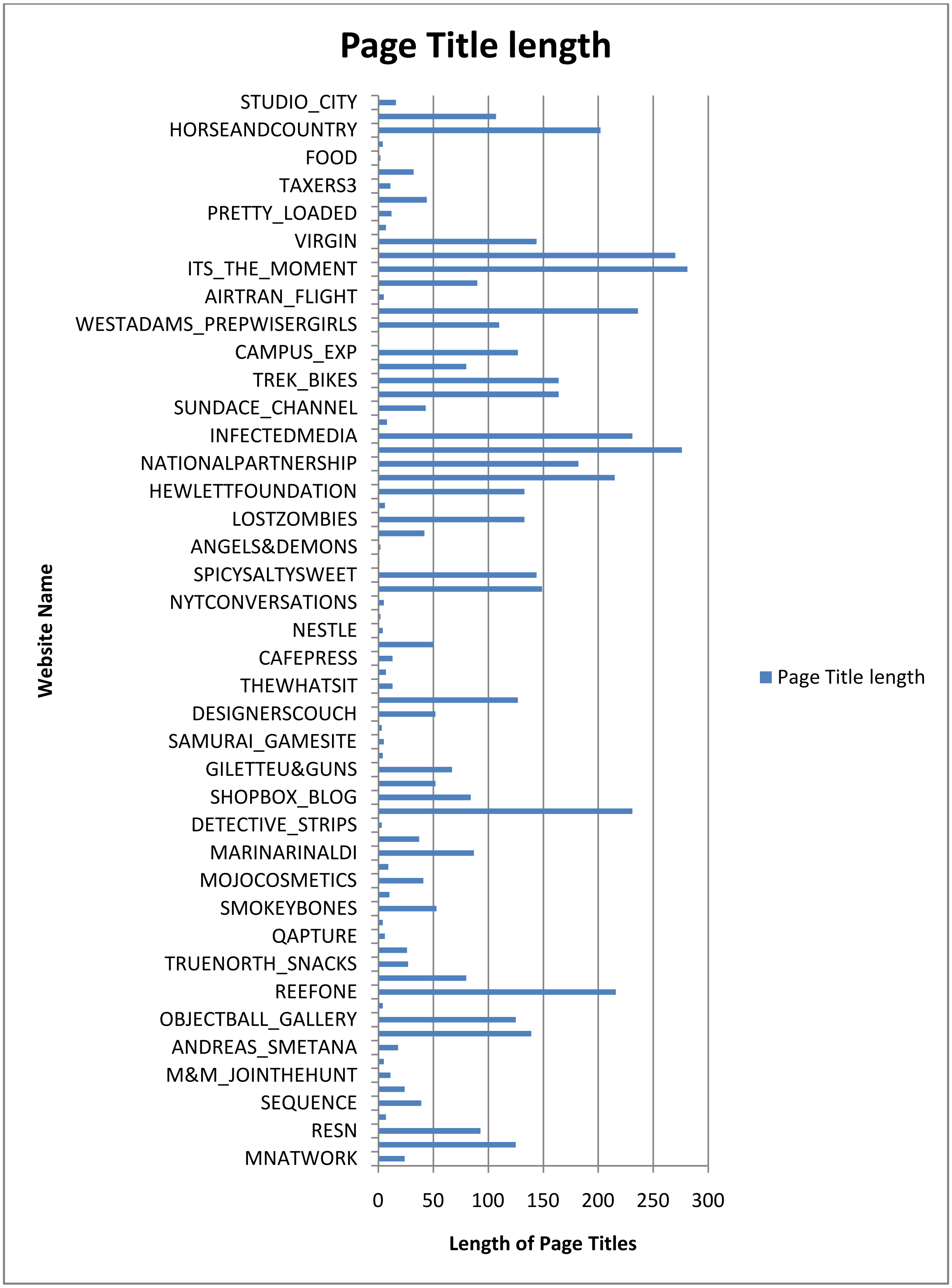

Figure 4.7: Page Title Length metric estimation for dataset of 2009

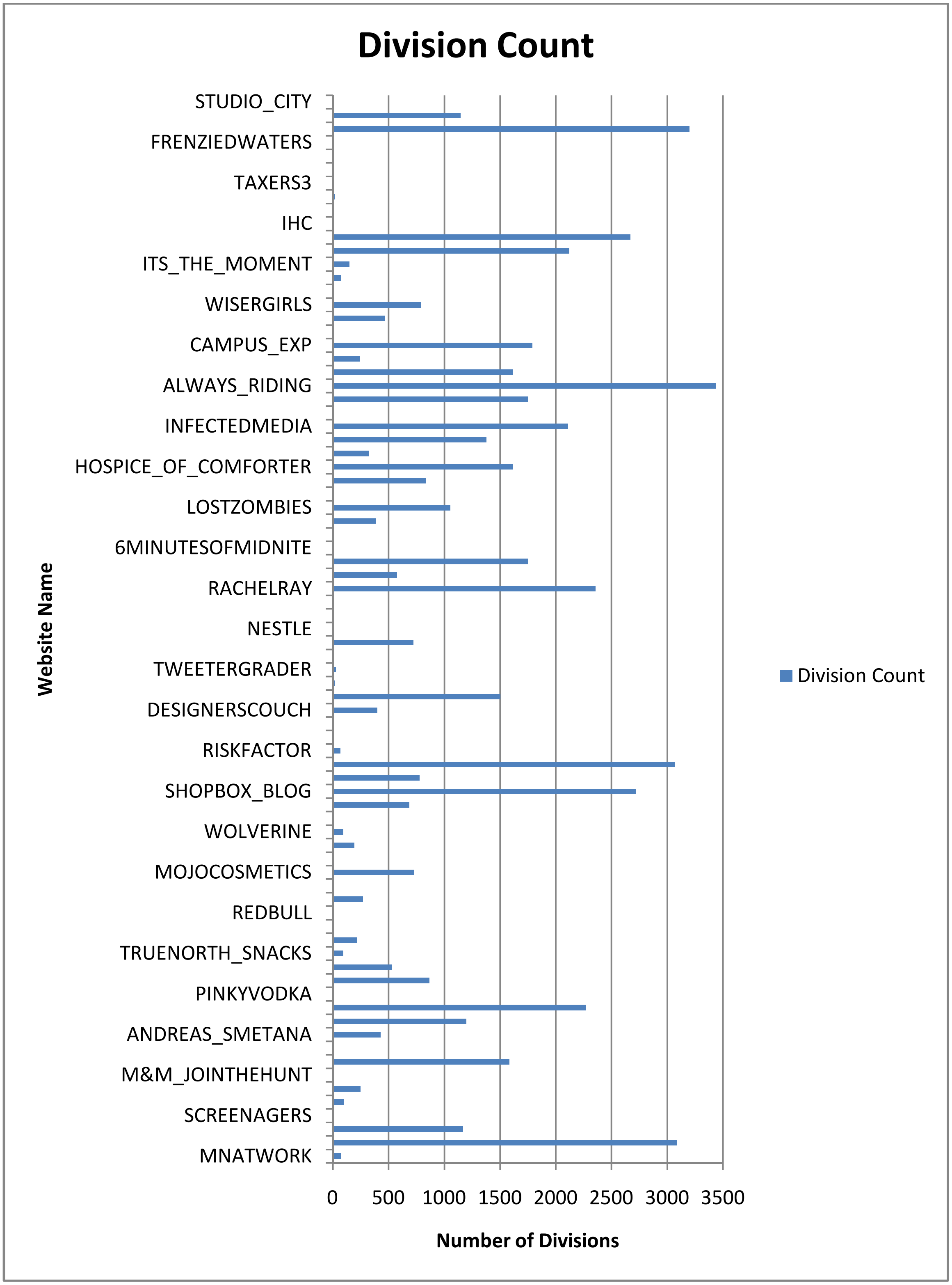

Figure 4.8: Division Count metric estimation for dataset of 2009

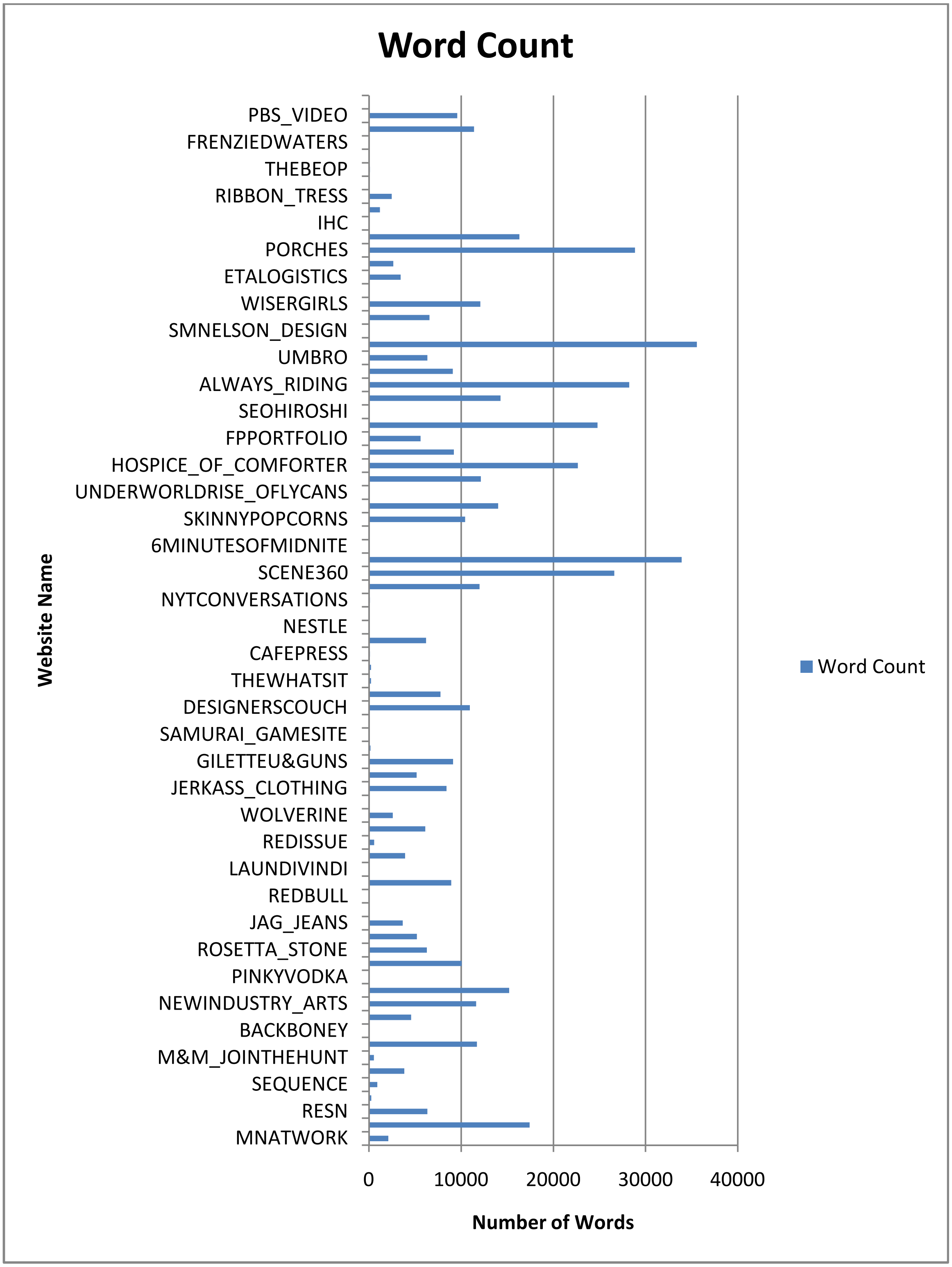

Figure 4.9: Word Count metric estimation for dataset of 2009

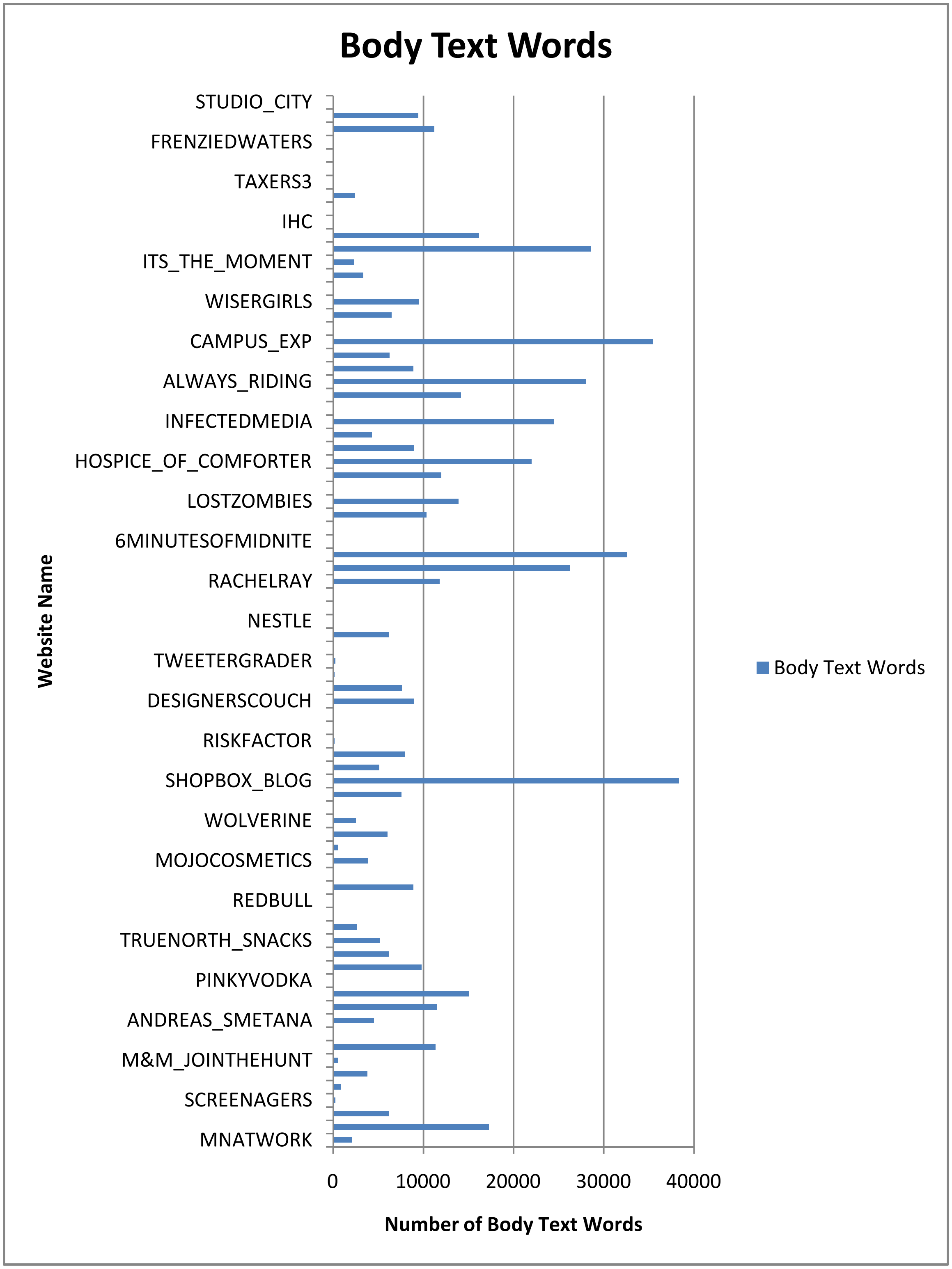

Figure 4.10: Body Text Words metric estimation for dataset of 2009

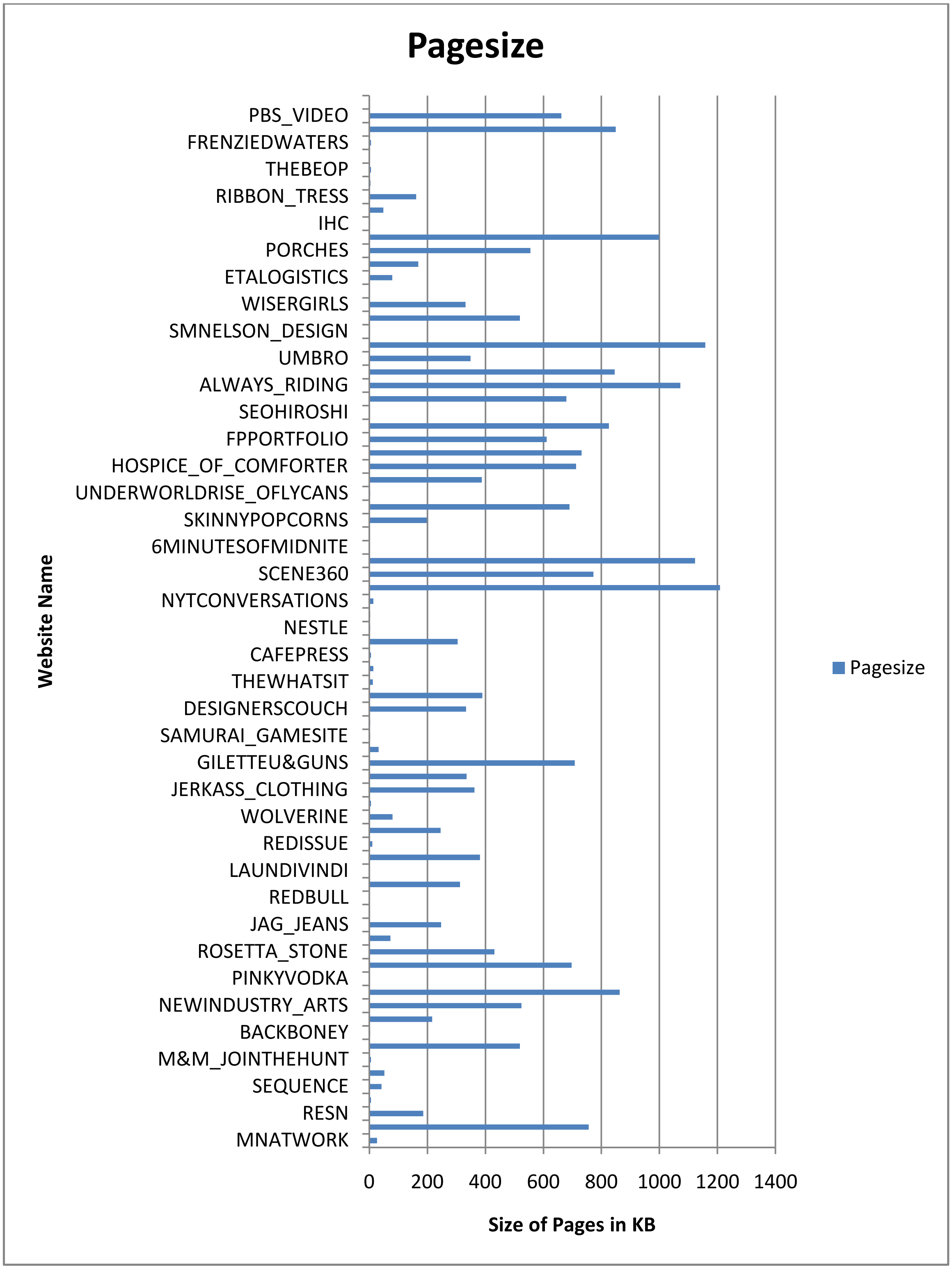

Figure 4.11: Pagesize metric estimation for dataset of 2009

4.3 Machine Learning Algorithms for Data Analysis

4.3.1 Multilayer Perceptron

A Multilayer Perceptron is a feed forward artificial neural network model that maps different input data instances onto a set of appropriate output. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Each node in all the layers is a neuron associated with a nonlinear activation function except for the input nodes. MLP utilizes a supervised learning technique called back-propagation for training the network. MLP is a modification of the standard linear perceptron, which can distinguish data that is not linearly separable [26]. Figure 4.12 shows the architecture of Multilayer Perceptron which contains one input layer, two hidden layers and one output layer.

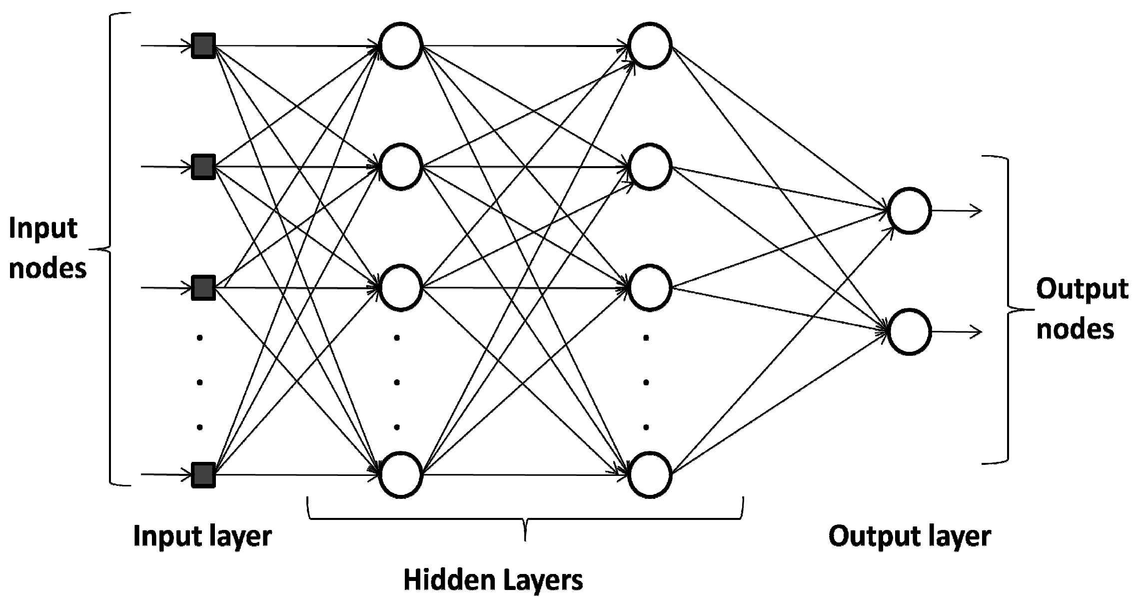

Figure 4.12: Architectural Graph of Multilayer Perceptron with two hidden layers

4.3.1.1 The Algorithm

The training of MLP proceeds in 2 phases:

1. In the forward phase, the synaptic weights are fixed and the values in the input pattern are propagated through the network layer by layer until it reaches the output.
2. In the backward phase, an error is generated by comparing the observed output of the network with the target response. The resulting error is propagated through the network, layer by layer in the backward direction. In this phase successive adjustments are applied to the synaptic weights.

4.3.1.2 Weight Training Calculation in Backward Phase

Let the input pattern be E. Let the target and observed response for node 'i' be $t_i(E)$ and oi(E) respectively. Let w_{ij} to specify weight between node i and node j

1. The Error Term for output unit k is calculated first as:

$$\delta_{O_k} = o_k(E)(1 - o_k(E))(t_k(E) - o_k(E))$$

2. The Error Term for hidden unit k is:

$$\delta_{H_k} = h_k(E)(1 - h_k(E)) \sum_{i\epsilon\ outputs} w_{ki}\delta_{O_i}$$

3. For each weight wij between input node i and hidden node j, calculate

$$\Delta_{ij} = \eta\delta_{H_j}x_i$$

where, x_i is the input to the network to the input node i for input pattern E and η is learning rate.

4. For each weight wij between hidden node i and output node j, calculate:

$$\Delta_{ij} = \eta\delta_{o_j}h_i(E)$$

where, hi(E) is the output from hidden node i for E.

5. Finally, add on each Δ_{ij} on to w_{ij}

$$w_{ij} = w_{ij} + \Delta_{ij}$$

6. In this way, the error is propagated back through the MLP.

4.3.2 Naive Bayes Classifier

Naïve Bayes classifier is a simple probabilistic classifier which is based on the Bayesian theorem which represents a supervised learning method. It naively assumes independence, it is only valid to multiply probabilities when the events are independent [29]. Given a class variable, a Naive Bayes classifier assumes that the presence of a particular feature of a class is not related to the presence of any other feature. Given the set of variables $X = \{x_1, x_2, x_3, \ldots\ x_n\}$, a probabilistic classifier can be defined as $p(C|x_1, x_2, x_3, \ldots\ x_n)$

Where, C is a dependent class variable with a set of possible outcomes conditional on several variables.

Using Bayes Theorem,

$$\mathrm{p(C)|p}(x_1, x_2, x_3, \ldots\ x_n) = \frac{\mathrm{p(C)\ p}(x_1, x_2, x_3, \ldots\ x_n\ |C)}{\mathrm{p}(x_1, x_2, x_3, \ldots\ x_n)}$$

Thus, we want to construct the posterior probability of the event C. Thus, the equation can be written as:

$$Posterior = \frac{Prior * likelihood}{Evidence}$$

Naïve Bayes algorithm is quite accurate and very fast and therefore, is a popular technique for classification. It is said that Naïve Bayes outperforms more sophisticated classifiers on many datasets, achieving impressive results [27].

4.3.3 Decision Tree

Decision tree learning uses a decision tree as a predictive model whose goal is to create a model that predicts the value of a target variable based on several input variables or attributes.

A Decision tree is a tree-structured plan of a set of attributes to test in order to predict the output [9]. In these tree structures, leaves represent class labels and branches represent conjunctions of attributes that lead to that class labels. C4.5 is one of the decision tree algorithms in the WEKA Tool [22] that we have used for our data analysis.

4.3.3.1 The Algorithm

C4.5 algorithm is used build the decision trees from a set of training data using the concept of information entropy.

1. Let the training data is a set $S = s_1,s_2,...$ of already classified instances. Each instance $\{s_i = x_1,x_2,...\}$ is a vector where $x_1,x_2,...$ represent attributes or features of the instance.
2. The training data is augmented with a vector $C = c_1,c_2,...$ where $c_1,c_2,...$ represent the class to which each instance belongs.
3. At each node of the tree, the algorithm chooses one attribute of the data that most effectively splits its set of instances into subsets enriched in one class or the other. Its criterion that results from choosing a best attribute for splitting the data is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen at each node to make the decision.
4. The algorithm then performs the whole procedure at each node.

This algorithm has a few cases to deal with:

1. All the instances in the dataset belong to the same class. When this happens, it creates a leaf node for the decision tree saying to choose that class.
2. None of the attribute provides any information gain. In this case, the algorithm creates a decision node higher up the tree using the expected value of the class.
3. Instance of previously-unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value.

4.3.4 Bagging

Bagging is an acronym for Bootstrap Aggregating. It was proposed by Leo Breiman [2] in 1994 to improve the classification by combining classifications of randomly generated training sets. It is a machine learning ensemble method to improve machine learning and statistical classification of regression models in terms of stability and classification accuracy. Bagging is a meta-algorithm which is based on averaging the results of various bootstrap samples. It is usually applied to decision tree models, but it can be used with any type of model.

Bagging = Bootstrapping + Aggregation

A learning set of L consists of data $\{(y_n, x_n), n = 1, 2, \ldots, N\}$ where the y's are either class labels or a numerical response. Bagging is a procedure for using this learning set to form a predictor $\varphi(\boldsymbol{x}, L)$ — if the input is $\boldsymbol{x}$ we predict y by $\varphi(\boldsymbol{x}, L)$.

4.3.4.1 Aggregation

Suppose we are given a sequence of learning sets $\{L_k\}$ each consisting of N independent observations from the same underlying distribution as L. Our mission is to use the $\{L_k\}$ to get a better predictor than the single learning set predictor $\varphi(\boldsymbol{x}, L)$. The restriction is that all we are allowed to work with is the sequence of predictors $\{\varphi(\boldsymbol{x}, L)\}$. If y is numerical, $\varphi(\boldsymbol{x}, L)$ is replaced by average of $\varphi(\boldsymbol{x}, L_k)$ over k.

$$\varphi_A(\boldsymbol{x}) = E_L\, \varphi(\boldsymbol{x}, L)$$

where E_L denotes the expectation over L, and the subscript A in φ_A denotes aggregation.

4.3.4.2 Bootstrapping

Though we have a single learning set L without the replicates of L. We sample the dataset with replacement to form the repeated bootstrap samples $\{L^{(B)}\}$ from L, and form $\{\varphi(\boldsymbol{x}, L^{(B)})\}$. If y is numerical, take φ_B as

$$\varphi_B(\boldsymbol{x}) = av_B\, \varphi(\boldsymbol{x}, L^{(B)})$$

Since some elements in this second dataset will be repeated, there must be some instances in the original dataset that have not been taken in the bootstrap sample: we use these instances as test instances.

Bagging provides a substantial reduction in prediction error for regression as well as classification methods. Since the method employs averaging of several predictors, it is not useful for improving linear models.

4.3.5 Random Forest

Random forest is an ensemble classifier that is made up of many decision trees and outputs the class that is the mode of the class's output by individual trees. The algorithm for inducing a random forest was developed by Leo Breiman and Adele Cutler in 1999. The term "Random Forest" came from "randomized decision forests" that was first proposed by Tin Kam Ho of Bell Labs in 1995. The method combines idea of bagging and the random selection of features, introduced independently by Ho and Amit and Geman in order to construct a collection of decision trees with controlled variation. Breiman [3] defines random forest as follows:

"A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k, k) = 1 \ldots\}$ *where the* Θ_k *are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input* x*."*

Random Forest grows many classification trees $(t_1, t_2 \ldots \ldots t_n)$ as shown in figure 4.13 and to classify a new object from an input vector V, the input vector is put down each of the trees in the forest. Each tree t_n gives a classification $P_n(c)$ for a particular category and we say the tree "votes" for that class. The forest chooses the classification having the most votes over all the trees in the forest.

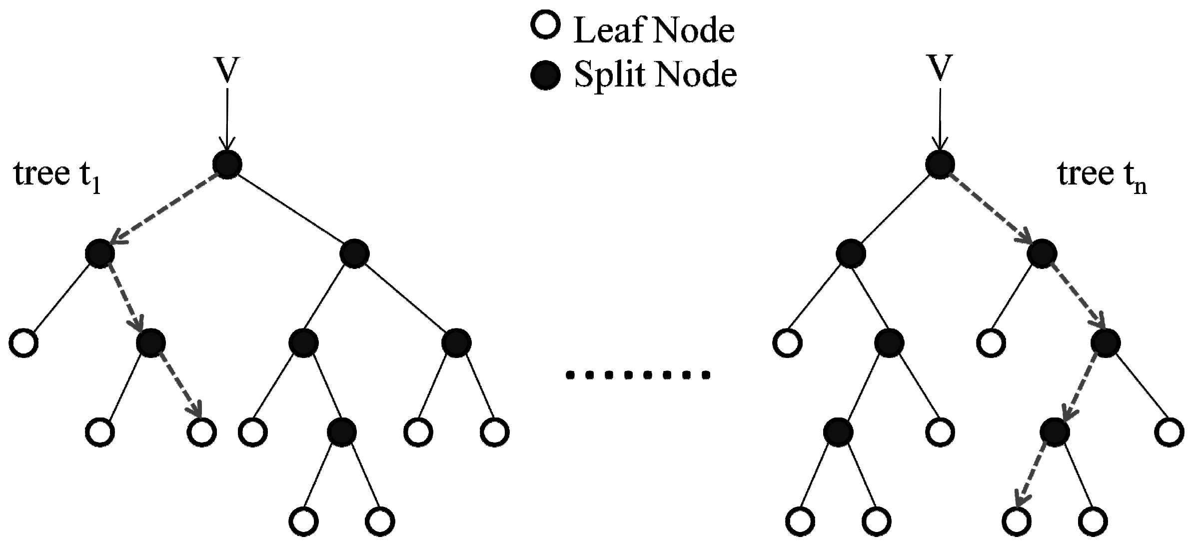

Figure 4.13: A Forest Of Trees

4.3.5.1 The Algorithm

The Random Forest algorithm for both classification and regression can be described as follows:

1. Choose T—number of trees to grow.
2. Choose m—number of variables used to split each node. m ≪ M, where M is the number of input variables. m is hold constant while growing the forest.
3. Grow T trees. When growing each tree do the following:

a. Construct a bootstrap sample of size n sampled from S_n with replacement and grow a tree from this bootstrap sample.

b. At each node, rather than choosing the best split among all predictor variables, select m variables at random and use them to find the best split.

4. Grow the tree to a maximal extent. There is no pruning.

   (Bagging: special case of random forests obtained when m, number of randomly sampled variables = M, total number of variables)

5. Predict the new data by aggregating the predictions of the T trees (i.e., majority votes for classification, average for regression).

In standard decision trees, each node is split on the basis of the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node. This counterintuitive strategy turns out to perform very well when compared to many other classifiers, including discriminant analysis, support vector machines and neural networks, and is robust against overfitting [30].

4.3.5.2 Out-Of-Bag (OOB) Error Estimation

There is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error. It is estimated internally. When the training set for the current tree is drawn by sampling with replacement, about one-third of the cases are left out of the sample. This is known as OOB data.

1. This OOB data is not used in the construction of k^{th} tree so we put each OOB case i.e. left out, down the k^{th} tree to get a classification. Thus, a test set classification is obtained for each case in about one-third of the trees.

2. At the end of the run, take 'J' to be the class predicted every time; case 'n' was OOB case. The proportion of times that 'J' is not equal to the true class of case 'n' - averaged over all cases is the OOB error estimate.

This has proven to be unbiased in many tests.

4.3.6 AdaBoost

AdaBoost, is an acronym for Adaptive Boosting, which is a machine learning algorithm, formulated by Yoav Freund and Robert Schapire in 1995. It is a meta-heuristic, and thus used in conjunction with many other learning algorithms to improve their performance [7]. It was the first algorithm that could adapt to the weak learners.

Boosting allows the use of weighted sample of training data to focus learning on difficult instances instead of taking the random sample and weighted vote instead of combining the classifiers with equal vote [28]. In boosting, we randomly select $n_1 < n$ samples from a given dataset D without replacement to obtain a new set of samples, which is used to train weak learner L_1. Next, we select $n_2 < n$ samples from D which includes half of the samples that were misclassified by learner L_1 and is used to train another weak learner C_2. Finally, we select all samples from dataset D, which were misclassified by both C_1 and C_2, and then we train C_3. The vote of all the weak learners is taken as final classifier.

4.3.6.1 The Algorithm

Given n training instances $(x_1, y_1), (x_2, y_2)..(x_n, y_n)$ where $x_i \in X$ and $y_i \in Y = \{-1, +1\}$.

Repeat for $t = 1..T$.

1. First, distribution is initialized over training set $D_1(i) = 1/m$.
2. Train learning algorithm on selected instances using distribution D_t to generate hypothesis h_t.
3. Choose a weight $\propto_t \in R$.
4. Update the distribution over the training set.
5. Then, a final vote $H(x)$ is taken as a weighted sum

$$H(x) = sign(f(x)) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$$

where $h_t(x)$ = weak or basis classifier/hypothesis and $H(x) =$ strong or final classifier/hypothesis.

4.3.7 Random Tree

Random tree is a single tree constructed in the Random Forest, or we can say that random forest is constructed by bagging ensembles of random trees. At each node of random tree, we select a given number of random features to find the best split and grow the tree to the maximum extent. There is no pruning.

4.3.8 Decision Table

Decision table is a non graphical way of showing the steps involved in taking a decision. The structure of decision table is made up of 4 quadrants which is shown in table 4.1.

Table 4.1: Structure of Decision Table

Conditions	Condition Alternative
Rules	Action Entries

Each condition is made from a decision question whose possible combinations with other decisions are listed in condition alternative area. Actions are the final outcomes of the decision process. They are represented as the branch ends in the decision tree. Action entries specify whether the action is to be performed for the set of condition alternatives the entry corresponds to.

Chapter 5

RESULTS ANALYSIS

In this chapter, we analyze the metrics estimations calculated for finding the relationship between web page measures and goodness of websites. To analyze the results of metrics on the dataset of 2678 web pages from 255 websites, we have employed 8 machine learning algorithms explained in the previous chapter, that are used to predict model best suited for evaluating the quality of websites. The following measures are used to evaluate the performance of each predicted model:

1. **Sensitivity and Specificity:** The sensitivity and specificity of the model are computed to predict the correctness of the model. The percentage of websites correctly predicted to be good is known as the sensitivity (True Positive Rate i.e. TPR) of the model. The percentage of websites correctly predicted to be bad is called specificity (1- False Positive Rate i.e. FPR) of the model. Ideally, both the sensitivity and specificity should be high to predict good and bad websites.

$$\text{Sensitivity (TPR)} = \frac{\text{True positives}}{\text{True positives} + \text{False Negatives}}$$

$$\text{Specificity } (1 - \text{FPR}) = \frac{\text{False positives}}{\text{True Negatives} + \text{False positives}}$$

Where, TP = websites correctly predicted to be good,

TN = websites correctly predicted to be bad,

FP = websites incorrectly predicted to good,

FN= websites incorrectly predicted to be bad.

2. **Receiver Operating Characteristic (ROC) analysis:** The performance of the outputs of the predicted models are evaluated using ROC analysis. It is an effective method of evaluating the quality or performance of predicted models.

The ROC curve is defined as a plot of sensitivity on the y-coordinate versus its (1-specificity) on the x coordinate. It is also known as a Relative Operating Characteristic curve, because it is a comparison of two operating characteristics (TPR & FPR). The construction of ROC curves allowed us to select many cutoff points between 0 and 1, and to calculate sensitivity and specificity at each cut off point. The optimal cutoff point that maximizes both sensitivity and specificity can be selected from the ROC curve. Hence, one can easily determine optimal cutoff point for a predicted model using the ROC curve.

The accuracy of the model can be predicted by applying it to the different data sets. We therefore, performed a 10-cross validation of the models [19]. Each dataset is randomly divided into 10 equal subsets. Each time one of the 10 subsets is used as the test set and the other 9 subsets are used to form a training set. Model I is predicted with respect to dataset of 2009. Model II is predicted with respect to dataset of 2010 and Model III is predicted with respect to dataset of 2011.

Section 5.1 shows the descriptive statistics; section 5.2 gives the analysis of logistic regression statistical technique and section 5.3-5.10 describe the analyses of machine learning techniques and finally section 5.11 discusses the evaluated results.

5.1 Descriptive Statistics

Descriptive statistics provide the simple summaries about the sample dataset and quantitatively describe its main features. Table 5.1- 5.3 presented below show “min”, “max”, “mean”, “median”, “std dev” for all the metrics in all the datasets 2009-2011.

Table 5.1: Descriptive Statistics of dataset 2009

	Min	Max	Mean	SD
Table Count	0	388	17.57	56.17
Link Count	0	4092	915.54	1151.57
Graphic Count	0	1130	180.73	265
List Count	0	706	93.57	158.96
Page Title Length	1	368	79.73	89.90
Division Count	0	3434	696.77	941.52
Word Count	1	41158	7233.41	9333.51
Body Text Words	0	38359	6980.84	9092.20
Page Size	1	1501	322.37	374.12

Table 5.2: Descriptive Statistics of dataset 2010

	Min	Max	Mean	SD
Table Count	0	570	14.58	65.74
Link Count	0	10619	809.32	1518.97
Graphic Count	0	1806	168.18	304.88
List Count	0	735	94.56	154.08
Page Title Length	0	333	62.04	85.23
Division Count	0	4581	574.20	921.21
Word Count	1	48964	6042.39	9167.66
Body Text Words	0	117720	7005.63	14855.95
Page Size	1	1817	276.53	410.78

Table 5.3: Descriptive Statistics of dataset 2011

	Min	Max	Mean	SD
Table Count	0	539	9.38	58.23
Link Count	1	3576	632.03	804.71
Graphic Count	0	1450	180.88	271.85
List Count	0	862	84.59	134.26
Page Title Length	0	323	56.50	61.39
Division Count	1	7254	538.76	938.17
Word Count	1	28205	5202.37	5932.24
Body Text Words	0	27882	4868.35	5798.27
Page Size	2	1108	252.43	282.80

5.2 Logistic Regression Analysis

Table 5.4 shows the web page prediction of logistic regression for all the 3 models. Table 5.5 shows the 10-cross validation results of all the 3 models.

Table 5.4: Website Prediction of Logistic Regression for Model I, II and III

Parameter	Model I	Model II	Model III
Number of good websites correctly predicted	15	24	20
Number of bad websites correctly predicted	49	54	46

Table 5.5: 10-cross validation results for models using Logistic Regression

	Sensitivity	Specificity	Cutoff	AUC
Model I	78.90	81.70	0.490	0.776
Model II	80.60	81.40	0.194	0.817
Model III	78.80	71.70	0.222	0.767

5.3 Multilayer Perceptron Analysis

The result of multilayer perceptron technique over the dataset of 3 years (2009-2011) is shown below. We have used only 1 hidden layer. There is only 1 output node in output layer whose value greater than a threshold (cutoff point) shows that category of website is good, otherwise it is not. The learning rate i.e. the amount the weights are updated is 0.9 and momentum applied to the weights during updating is 0.9. Table 5.6 shows the web page prediction of multilayer perceptron for all the 3 models. Table 5.7 shows the 10-cross validation results of all the 3 models.

The following observations are made from the analysis shown in table 5.6 and 5.7:

- Out of 19 good websites, only 7 websites are correctly predicted as good and 55 out of 60 websites are correctly predicted as bad which gives a sensitivity of 36.80% and 91.70%, respectively, which gives the weighted average sensitivity of 78.90%. 12 websites are incorrectly predicted as bad in model I which gives a FPR of 63.20% and 5 websites are incorrectly predicted as bad giving a FPR of 8.30%, and thus, a weighted average specificity (1-FPR) of 55.00%.
- Same applies for model II and model III.

Table 5.6: Website Prediction of Multilayer Perceptron for Model I, II and III

Parameter	Model I	Model II	Model III
Number of good websites correctly predicted	7	7	15
Number of bad websites correctly predicted	55	54	48

Table 5.7: 10-cross validation results for models using Multilayer Perceptron

	Sensitivity	Specificity	Cutoff	AUC
Model I	78.90	55.00	0.058	0.754
Model II	61.30	50.80	0.057	0.566
Model III	63.60	66.00	0.436	0.703

The ROC curves for multilayer perceptron models are shown in figure 5.1- 5.3.

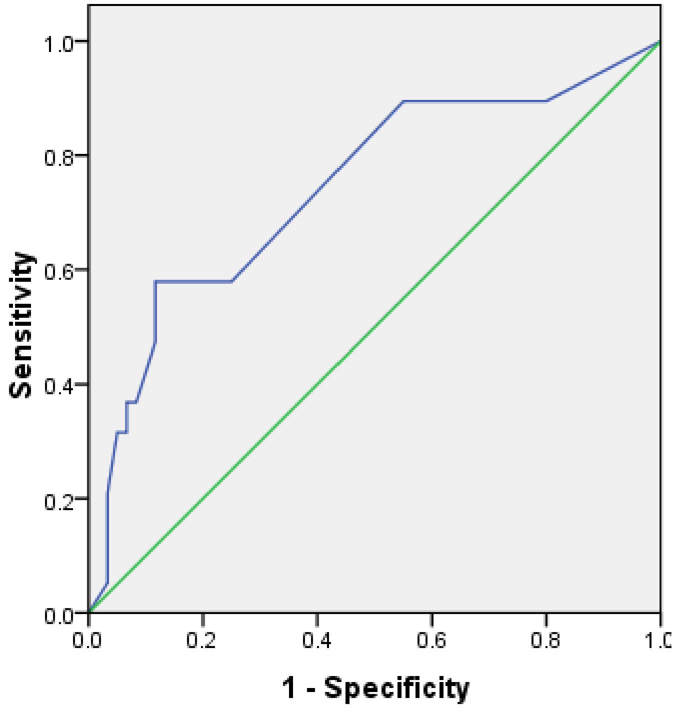

Figure 5.1: ROC Curve for Model I using Multilayer Perceptron

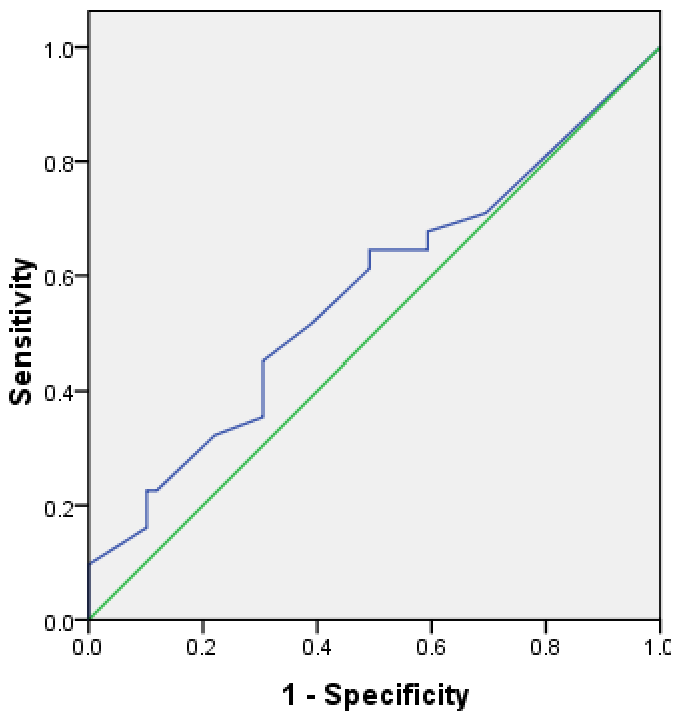

Figure 5.2: ROC Curve for Model II using Multilayer Perceptron

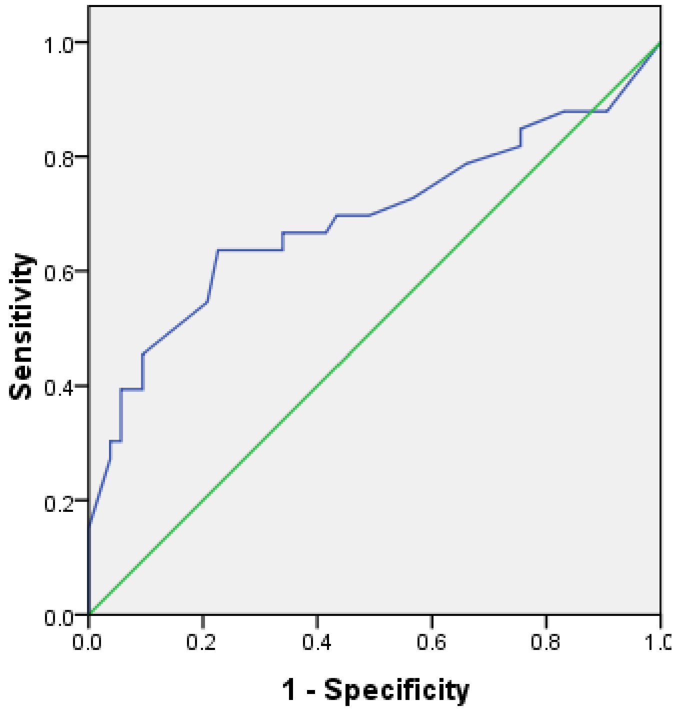

Figure 5.3: ROC Curve for Model III using Multilayer Perceptron

5.4 Naïve Bayes Analysis

Table 5.8 shows the web page prediction of Naïve Bayes classifier for all the 3 models. Table 5.9 shows the 10-cross validation results of all the 3 models.

The following observations are made from the analysis shown in table 5.8 and 5.9:

- In model I, out of 19 good websites, 18 websites are correctly predicted as good and out of 60 websites, 47 websites are correctly predicted as bad, which gives the sensitivity of 94.70% and specificity of 80.00%
- Similarly, in model II, 20 good websites are correctly predicted and rest are incorrectly predicted to be bad out of 31 while, 54 are correctly predicted to be bad, giving the sensitivity of 83.90% and specificity of 84.70%.

- Table 5.9 shows the sensitivity and specificity of model III to be 78.80% and 77.44%, respectively, as 20 out of 33 websites are correctly predicted to be good and 46 out of 53 websites are correctly predicted to be bad.

Table 5.8: Website Prediction of Naïve Bayes Classifier for Model I, II and III

Parameter	Model I	Model II	Model III
Number of good websites correctly predicted	18	20	20
Number of bad websites correctly predicted	47	54	46

Table 5.9: 10-cross validation results for models using Naïve Bayes classifier

	Sensitivity	Specificity	Cutoff	AUC
Model I	94.70	80.00	0.916	0.923
Model II	83.90	84.70	0.001	0.871
Model III	78.80	77.40	0.075	0.843

Figure 5.4 - 5.6 shows the ROC curves for all the models using Naïve Bayes classifier.

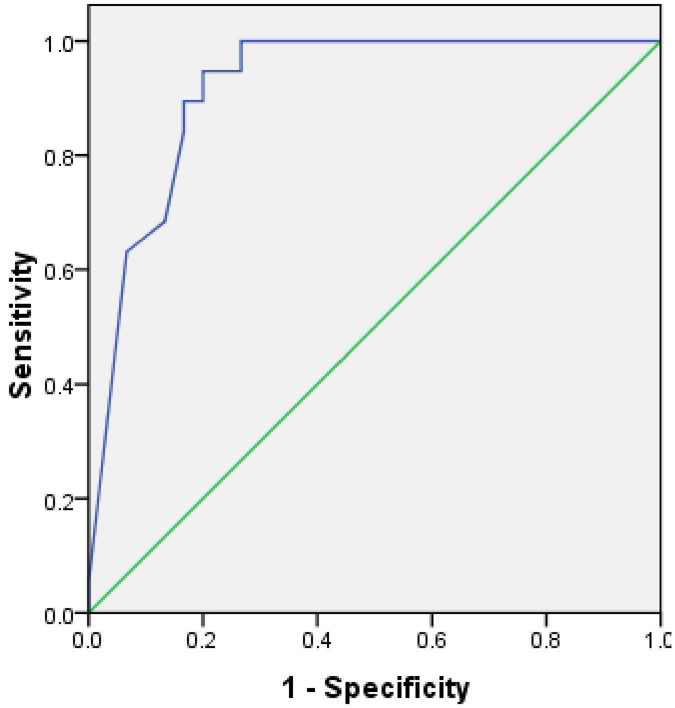

Figure 5.4: ROC Curve for Model I using Naïve Bayes Classifier

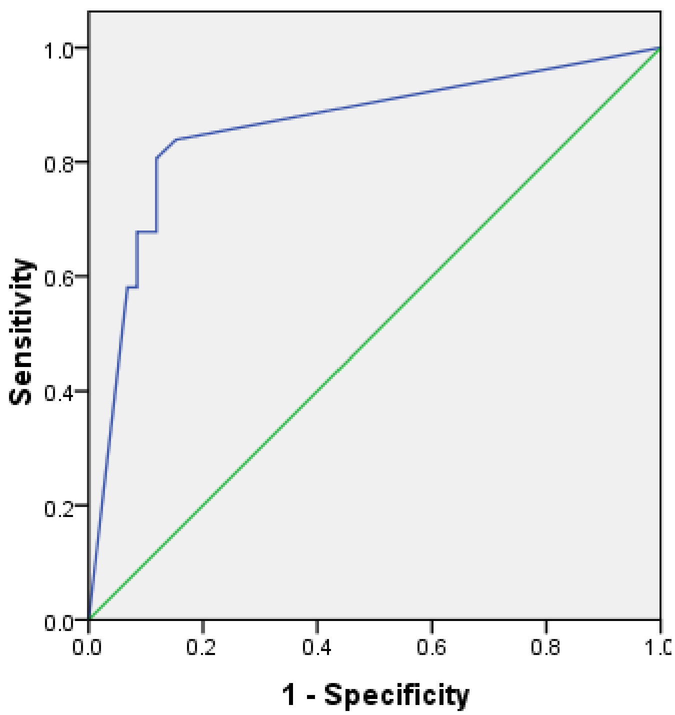

Figure 5.5: ROC Curve for Model II using Naïve Bayes Classifier

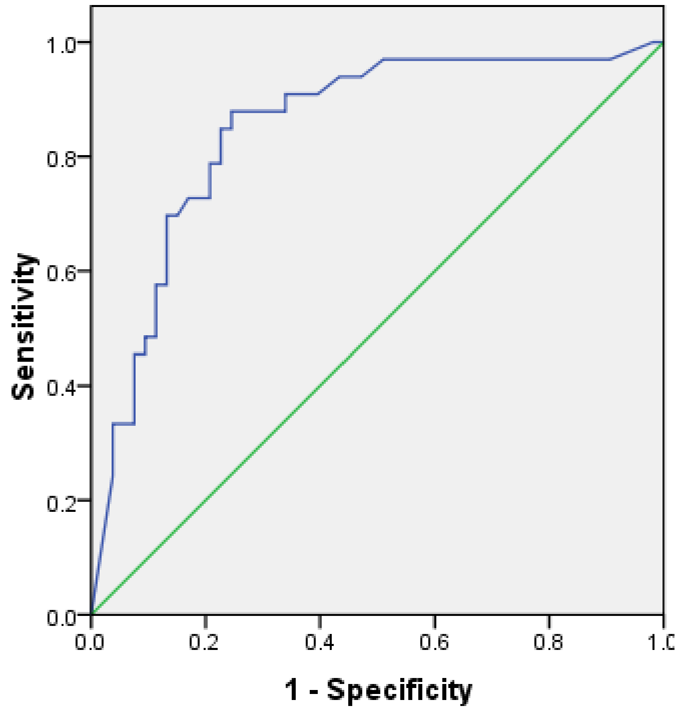

Figure 5.6: ROC Curve for Model III using Naïve Bayes Classifier

5.5 Decision Tree Analysis

In the Decision Tree method, an independent variable is selected at each node of the tree. The tree is traversed during classification from the root until a leaf node is reached. Each leaf node is associated with a decision or classification. C4.5 algorithm is used to create the decision tree. There are 5 leaves and tree size 9 in model I. Model II has 2 leaves and tree size of 3. Number of leaves in model III is 5 and size of the tree is 9. Table 5.10 shows the web page prediction of decision tree for all the 3 models. Table 5.11 shows the 10-cross validation results of all the 3 models.

The following observations are made from the analysis shown in table 5.10 and 5.11:

- Out of 19 good websites, 15 are correctly predicted and 49 out of 60 websites are correctly predicted as bad which gives a sensitivity of 89.50% and specificity of 81.70%, respectively.
- Same applies for model II and model III.

Table 5.10: Website Prediction of Decision Tree for Model I, II and III

Parameter	**Model I**	**Model II**	**Model III**
Number of good websites correctly predicted	15	21	24
Number of bad websites correctly predicted	49	51	41

Table 5.11: 10-cross validation results for models using Decision Tree

	Sensitivity	**Specificity**	**Cutoff**	**AUC**
Model I	89.50	81.70	0.154	0.829
Model II	71.00	76.30	0.124	0.755
Model III	81.80	71.70	0.295	0.777

The ROC curves for decision tree models are shown in figure 5.7- 5.9.

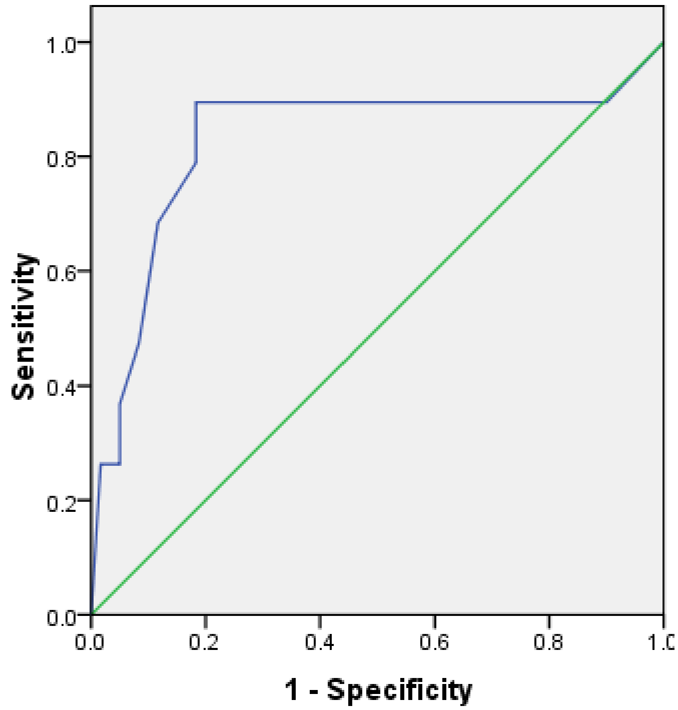

Figure 5.7: ROC Curve for Model I using Decision Tree

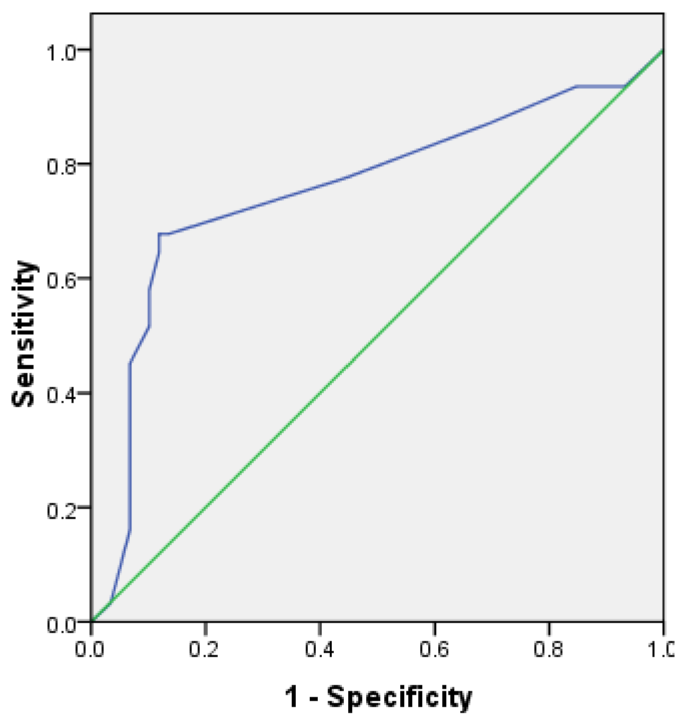

Figure 5.8: ROC Curve for Model II using Decision Tree

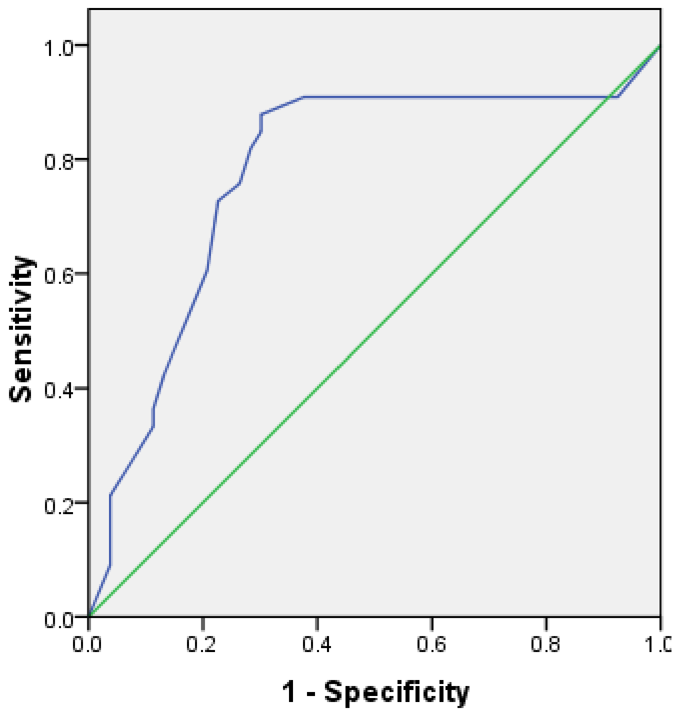

Figure 5.9: ROC Curve for Model III using Decision Tree

5.6 Bagging Analysis

We have used IB1 algorithm in bagging method present in WEKA tool. Number of iterations used are 10. Out of bag error in model I, model II and model III is 0.241, 0.278 & 0.337, respectively. Table 5.12 shows the web page prediction of bagging for all the 3 models. Table 5.13 shows the 10-cross validation results of all the 3 models.

The following observations are made from the analysis shown in table 5.12 and 5.13:

- Out of 19 good websites, 12 are correctly predicted and 51 out of 60 websites are correctly predicted as bad which gives a sensitivity of 78.90% and specificity of 75.00%, respectively.
- Same applies for model II and model III.

Table 5.12: Web page Prediction of Bagging for Model I, II and III

Parameter	Model I	Model II	Model III
Number of good websites correctly predicted	12	19	22
Number of bad websites correctly predicted	51	45	37

Table 5.13: 10-cross validation results for models using Bagging

	Sensitivity	Specificity	Cutoff	AUC
Model I	78.90	75.00	0.050	0.806
Model II	71.00	69.50	0.150	0.77
Model III	66.70	71.70	0.650	0.708

Figure 5.10- 5.12 shows the ROC curves for all the 3 models using bagging.

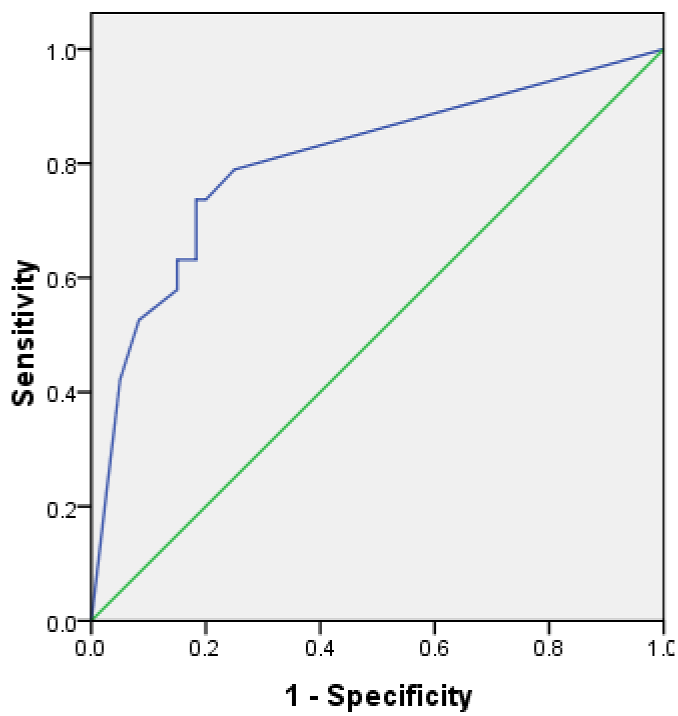

Figure 5.10: ROC Curve for Model I using Bagging

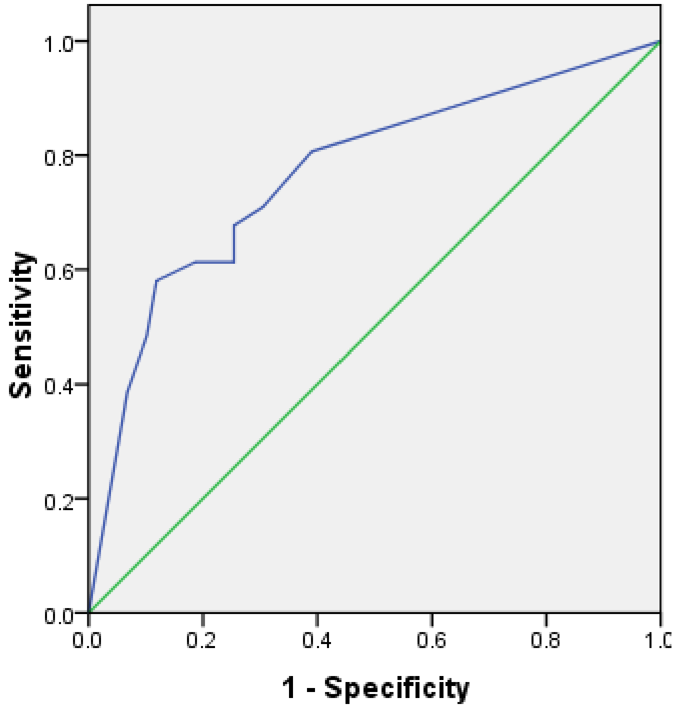

Figure 5.11: ROC Curve for Model II using Bagging

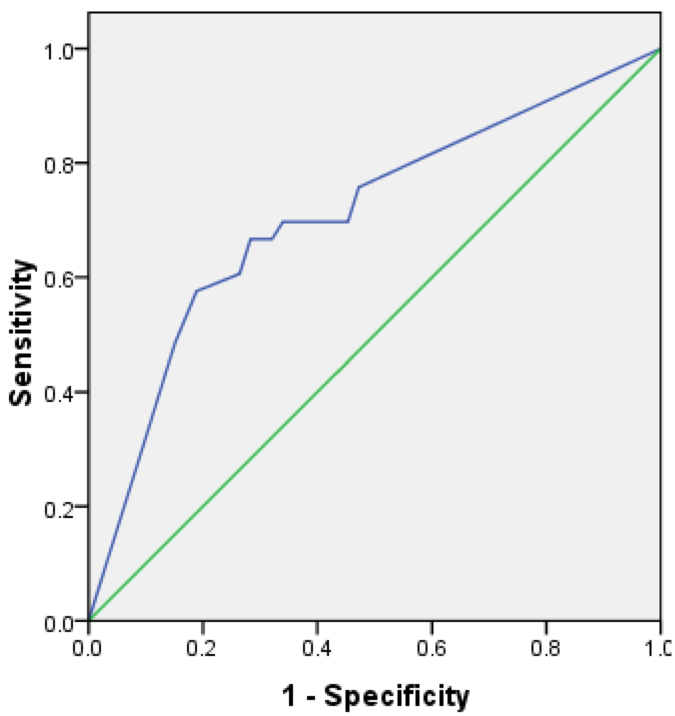

Figure 5.12: ROC Curve for Model III using Bagging

5.7 Random Forest Analysis

For each of the model, random forest of 10 trees is constructed and each constructed while considering 4 random independent variables at each node. In model I, out of bag error is found to be 0.190. Out of bag error in model II and model III are 0.233 and 0.302, respectively. Table 5.14 shows the web page prediction of random forest for all the 3 models. Table 5.15 shows the 10-cross validation results of all the 3 models.

The following observations are made from the analysis shown in table 5.14 and 5.15:

- In model I, out of 19 good websites, 13 websites are predicted good and out of 60 websites, 54 websites are correctly predicted as bad, which gives the sensitivity of 78.90% and specificity of 80.00%
- Similarly, in model II, 18 good websites are correctly predicted and rest are incorrectly predicted to be bad out of 31 while, 50 are correctly predicted to be bad, giving the sensitivity of 80.60% and specificity of 78.00%.
- Table 5.15 shows the sensitivity and specificity of model III to be 75.80% and 67.90%, respectively, as 22 out of 33 websites are correctly predicted to be good and 40 out of 53 websites are correctly predicted to be bad.

Table 5.14: Website Prediction of Random Forest for Model I, II and III

Parameter	**Model I**	**Model II**	**Model III**
Number of good websites correctly predicted	13	18	22
Number of bad websites correctly predicted	54	50	40

Table 5.15: 10-cross validation results for models using Random Forest

	Sensitivity	Specificity	Cutoff	AUC
Model I	78.90	80.00	0.150	0.843
Model II	80.60	78.00	0.284	0.799
Model III	75.80	67.90	0.250	0.787

Figure 5.13- 5.15 shows the ROC curves for all the 3 models using random forest.

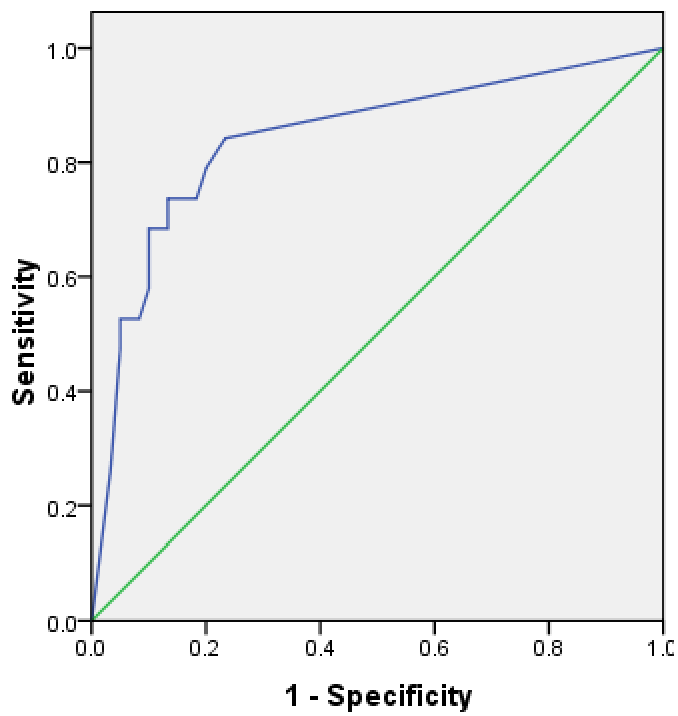

Figure 5.13: ROC Curve for Model I using Random Forest

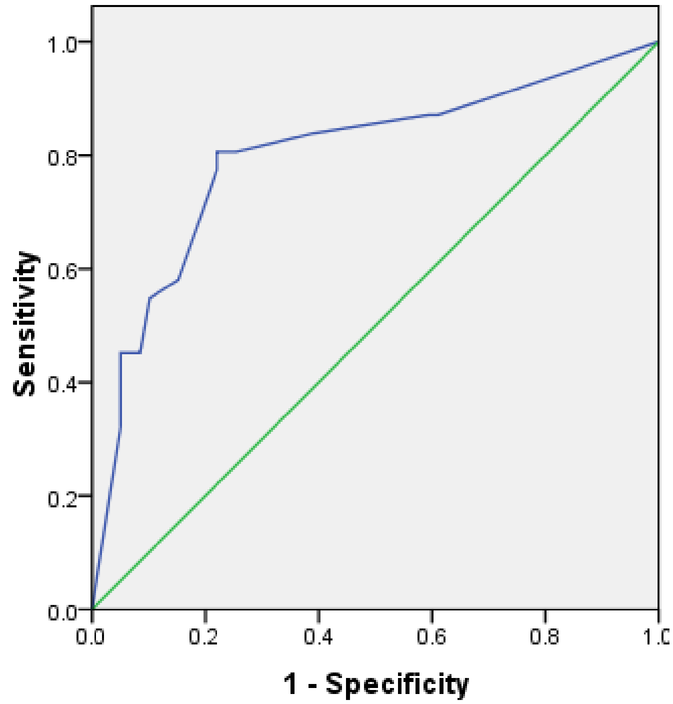

Figure 5.14: ROC Curve for Model II using Random Forest

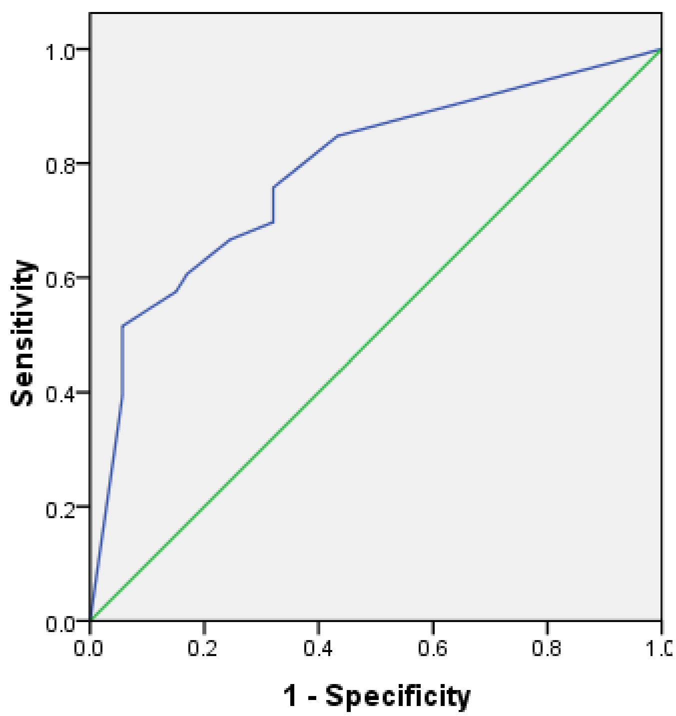

Figure 5.15: ROC Curve for Model III using Random Forest

5.8 AdaBoost Analysis

We have used REPTree algorithm implemented in WEKA tool. There are 6 iterations that are performed for each model. Table 5.16 shows the web page prediction of random forest for all the 3 models. Table 5.17 shows the 10-cross validation results of all the 3 models.

The following observations are made from the analysis shown in table 5.16 and 5.17:

- Out of 19 good websites, 11 are correctly predicted and 51 out of 60 websites are correctly predicted as bad which gives a sensitivity of 78.90% and specificity of 81.10%, respectively.
- Same applies for model II and model III.

Table 5.16: Website Prediction of AdaBoost for Model I, II and III

Parameter	**Model I**	**Model II**	**Model III**
Number of good websites correctly predicted	11	19	25
Number of bad websites correctly predicted	51	52	40

Table 5.17: 10-cross validation results for models using AdaBoost

	Sensitivity	**Specificity**	**Cutoff**	**AUC**
Model I	78.90	81.10	0.137	0.839
Model II	64.50	86.40	0.227	0.720
Model III	78.80	74.20	0.436	0.808

Figure 5.16- 5.18 shows the ROC curves for all the 3 models using AdaBoost.

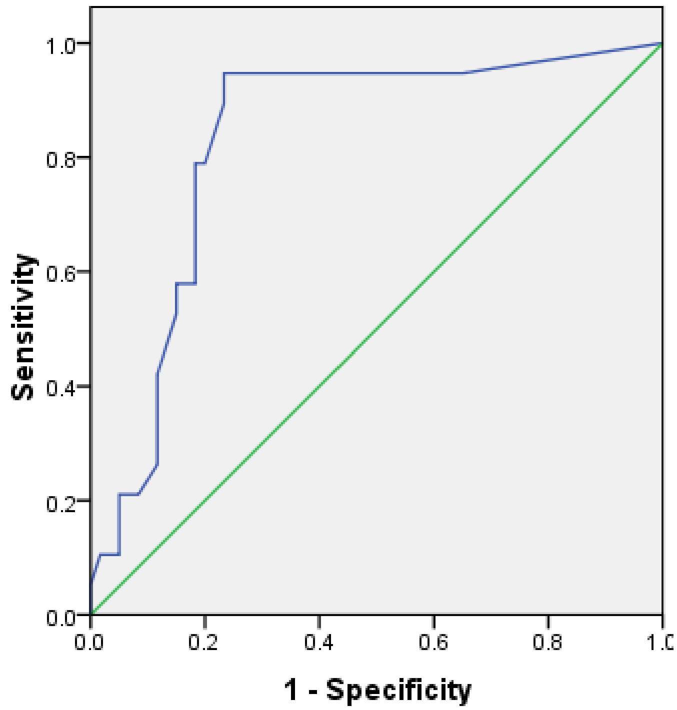

Figure 5.16: ROC Curve for Model I using AdaBoost

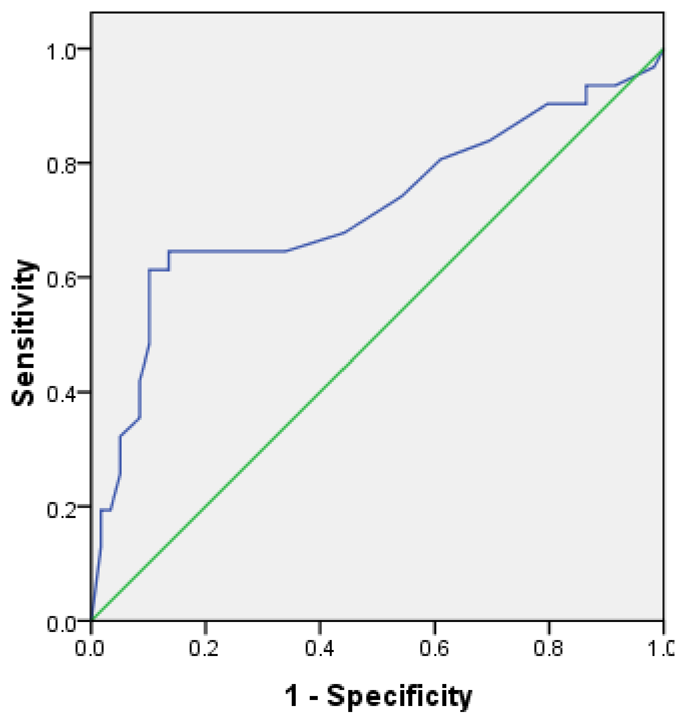

Figure 5.17: ROC Curve for Model II using AdaBoost

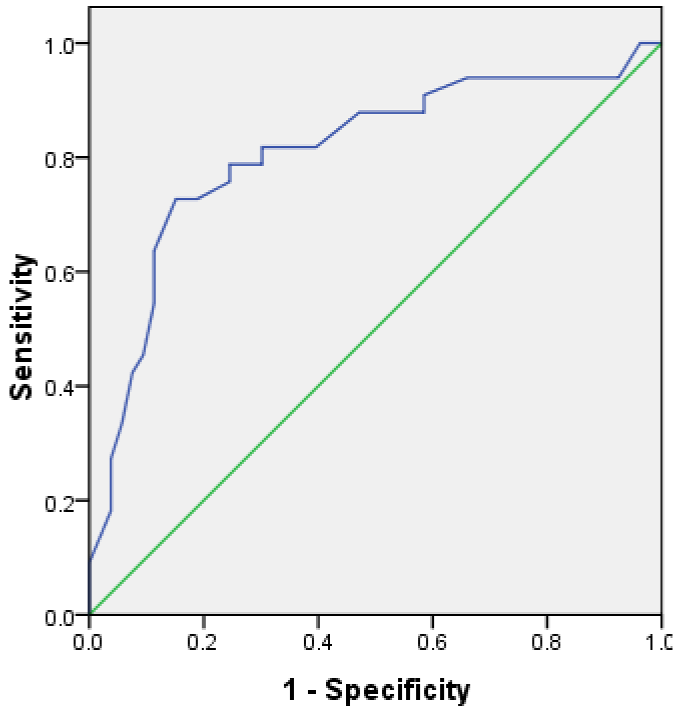

Figure 5.18: ROC Curve for Model III using AdaBoost

5.9 Random Tree Analysis

The tree size in model I, model II and model III are 27, 47 & 39, respectively. Table 5.18 shows the web page prediction of random tree for all the 3 models. Table 5.19 shows the 10-cross validation results of all the 3 models.

The following observations are made from the analysis shown in table 5.18 and 5.19:

- Out of 19 good websites, 12 are correctly predicted and 52 out of 60 websites are correctly predicted as bad which gives a sensitivity of 73.70% and specificity of 86.70%, respectively.
- Same applies for model II and model III.

Table 5.18: Website Prediction of Random Tree for Model I, II and III

Parameter	Model I	Model II	Model III
Number of good websites correctly predicted	12	20	20
Number of bad websites correctly predicted	52	45	39

Table 5.19: 10-cross validation results for models using Random Tree

	Sensitivity	Specificity	Cutoff	AUC
Model I	73.70	86.70	0.250	0.791
Model II	64.50	76.30	0.750	0.695
Model III	60.60	73.60	0.500	0.671

Figure 5.19- 5.21 shows the ROC curves for all the 3 models using Random Tree.

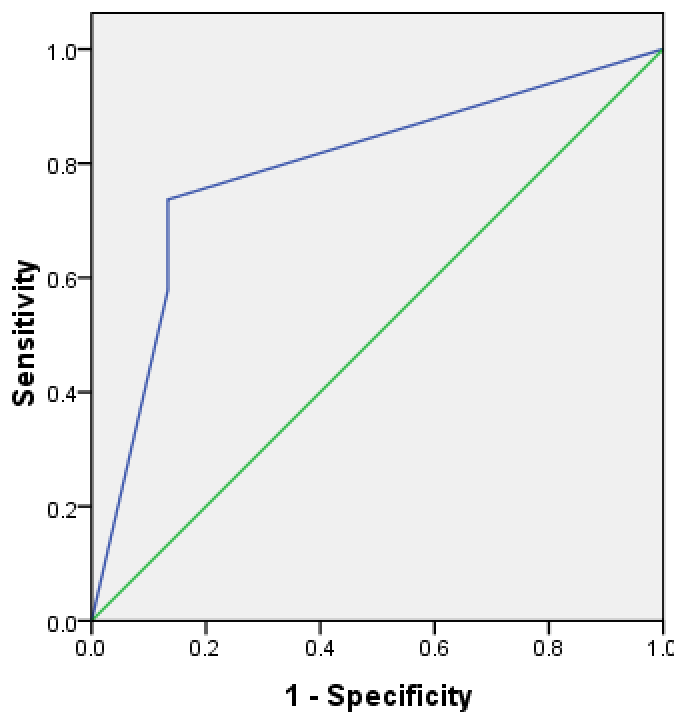

Figure 5.19: ROC Curve for Model I using Random Tree

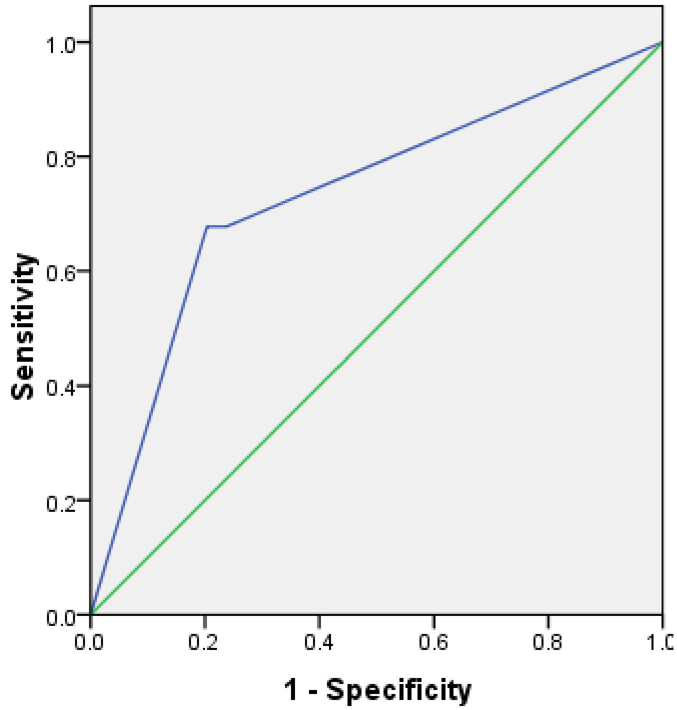

Figure 5.20: ROC Curve for Model II using Random Tree

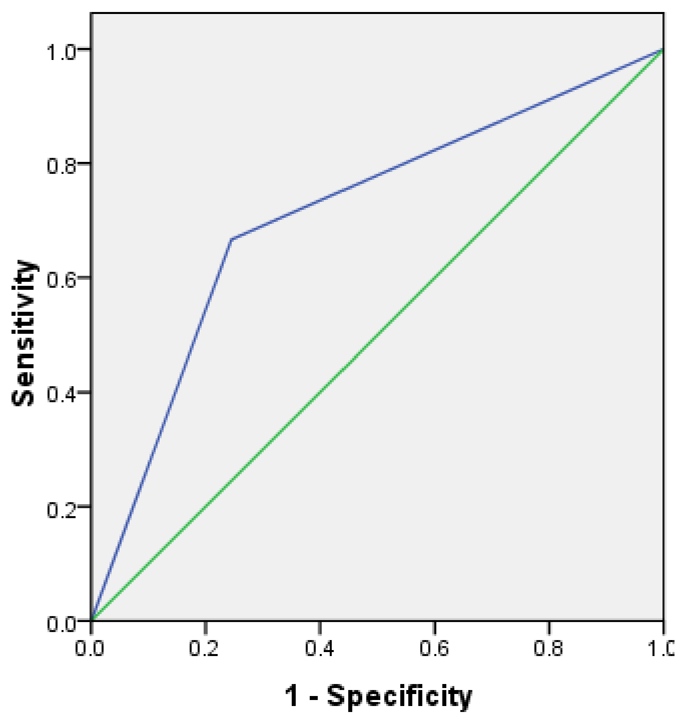

Figure 5.21: ROC Curve for Model III using Random Tree

5.10 Decision Table Analysis

It uses a best first search technique. Table 5.20 shows the web page prediction of random forest for all the 3 models. Table 5.21 shows the 10-cross validation results of all the 3 models.

The following observations are made from the analysis shown in table 5.20 and 5.21:

- Out of 19 good websites, 17 are correctly predicted and 49 out of 60 websites are correctly predicted as bad which gives a sensitivity of 94.70% and specificity of 81.70%, respectively.
- Same applies for model II and model III.

Table 5.20: Website Prediction of Decision Table for Model I, II and III

Parameter	**Model I**	**Model II**	**Model III**
Number of good websites correctly predicted	17	20	23
Number of bad websites correctly predicted	49	52	43

Table 5.21: 10-cross validation results for models using Decision Table

	Sensitivity	**Specificity**	**Cutoff**	**AUC**
Model I	94.70	81.70	0.209	0.839
Model II	67.70	79.70	0.151	0.744
Model III	72.70	75.50	0.458	0.747

The ROC curves for all the 3 models are shown in figure 5.22 -5.24.

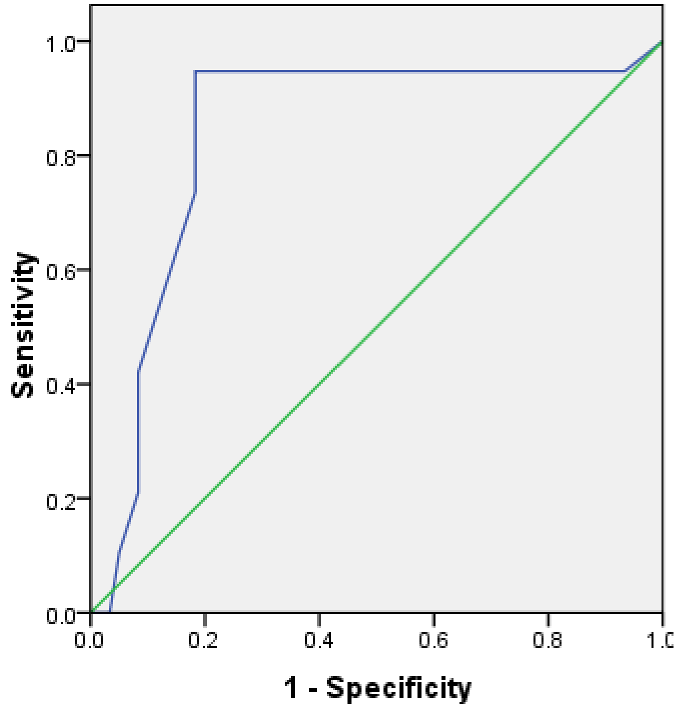

Figure 5.22: ROC Curve for Model I using Decision table

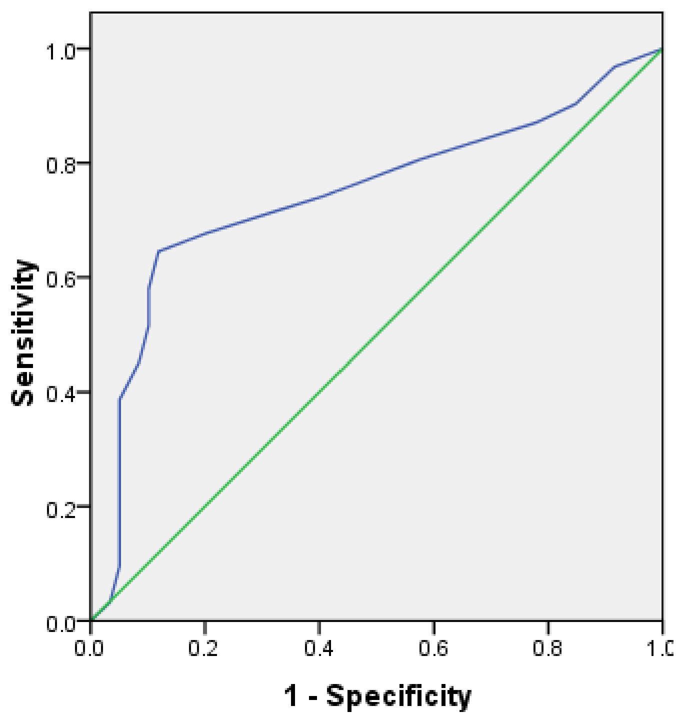

Figure 5.23: ROC Curve for Model II using Decision table

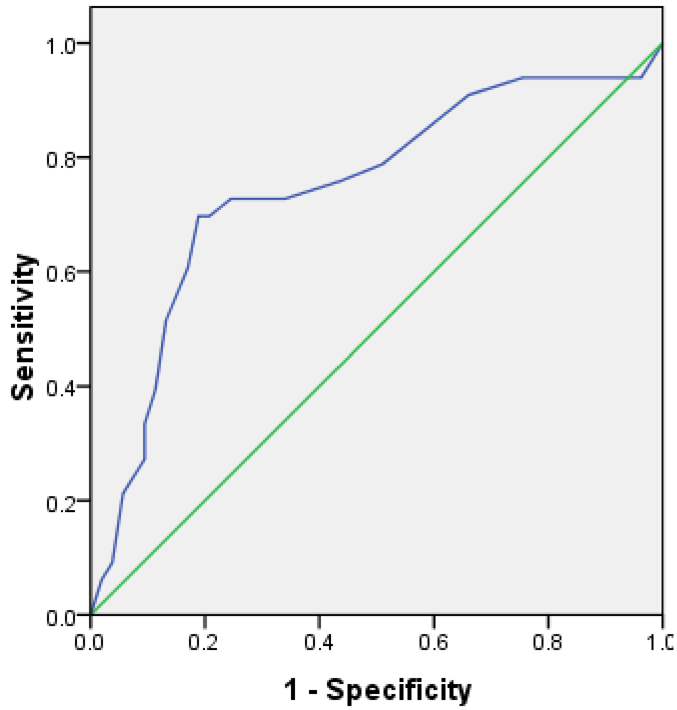

Figure 5.24: ROC Curve for Model III using Decision Table

5.11 Model Evaluation

We have used CFS technique to reduce the data dimensionality and thus, to select a subset of attributes explained in chapter 3. After applying CFS, the 9 variables were reduced to 3 variables (one dependent and 2 independent variables) when applied to dataset of 2009. The 2 independent variables are Link Count & Page Title Length. There are 5 variables selected in dataset of 2010 where the independent ones are Link Count, List Count, Word Count and Page Size. 6 significant variables which include Link Count, List Count, Page title Length, Word Count and Body Text Words as independent ones are selected in dataset of 2011.

Following inferences can be made from the selected significant metrics:

- Link count is attributed as significant metric in all the three datasets which makes it important to be considered by the designers. It shows that number of links should be appropriate to enhance the quality of a website.
- With each year, the number of significant metrics has also increased. List Count and Word Count are common for 2 year dataset.

We have not selected an arbitrary cutoff point and in order to obtain a balance between the number of websites predicted as good and bad, the cutoff point of the predicted model is computed using ROC analysis. Area under the ROC Curve (AUC) is a combined measure of sensitivity and specificity. Thus, we have used the area under the ROC curve for computing the accuracy of the predicted models. The models are applied on the same dataset from which they are derived using 10-cross validation of all the models. Table 5.22 summarizes the results of the analysis of the 8 machine learning techniques on all the 3 models.

Table 5.22 (a): Summarized result of 10-cross validation of all the models

	Multilayer Perceptron			**Naïve Bayes Classifier**			**Decision Tree**			**Bagging**		
	Model I	Model II	Model III	Model I	Model II	Model III	Model I	Model II	Model III	Model I	Model II	Model III
Sensitivity	78.90	61.30	63.60	94.70	83.90	78.80	89.50	71.00	81.80	78.90	71.00	66.70
Specificity	55.00	50.80	66..00	80.00	84.70	77.40	81.70	76.30	71.70	75.00	69.50	71.70
Cutoff	0.058	0.057	0.436	0.916	0.001	0.075	0.154	0.124	0.295	0.050	0.150	0.650
AUC	0.754	0.566	0.703	0.923	0.871	0.843	0.829	0.755	0.777	0.806	0.770	0.708

Table 5.22 (b): Summarized result of 10-cross validation of all the models

	Random Forest			AdaBoost			Random Tree			Decision Table		
	Model I	Model II	Model III	Model I	Model II	Model III	Model I	Model II	Model III	Model I	Model II	Model II
Sensitivity	78.90	80.60	75.80	78.90	64.50	78.80	73.70	64.50	60.60	94.70	67.70	72.70
Specificity	80.00	78.00	67.90	81.10	86.40	74.20	86.70	76.30	73.60	81.70	79.70	75.50
Cutoff	0.150	0.284	0.250	0.137	0.227	0.436	0.250	0.750	0.500	0.209	0.151	0.458
AUC	0.843	0.799	0.787	0.839	0.720	0.808	0.791	0.695	0.671	0.839	0.744	0.747

We have employed logistic regression and machine learning techniques to evaluate their performance for predicting the quality of the websites. The AUC of all the models predicted using Naïve Bayes technique is greater than the AUC of all the other models predicted using the logistic regression as well as other machine learning techniques (Multilayer Perceptron, Decision Tree, Bagging, Random Forest Adaboost, Random Tree and Decision Table). The model I with respect to dataset of 2009 has an AUC of 0.923 using Naïve Bayes technique which is greater than that using other techniques and same trend is seen for the models with respect to dataset of year 2010 and 2011 with the AUC of 0.871 and 0.843 respectively. All the models performed best with Naïve Bayes classifier, which is reflected in their AUC values.

Both the sensitivity and specificity should be high to predict good and bad websites. The models predicted with the Naïve Bayes technique have higher prediction performance in terms of sensitivity and specificity. For Model I, Naïve Bayes classifier provides the sensitivity of 94.70 and specificity of 80.00. Model II has sensitivity of 83.90 and specificity of 84.70. For Model III, Naïve Bayes provides the sensitivity and specificity of 78.80 and 77.40, respectively.

Thus, on overall basis in terms of sensitivity, specificity and area under ROC curve, the best model suitable for predicting the class of websites as good or bad is determined to be Naïve Bayes Model. It is said that Naïve Bayes outperforms more sophisticated classifiers on many datasets, achieving impressive results.

Chapter 6

CONCLUSIONS AND FUTURE WORK

The goal of the research is to find the effect of web page measures on the categorization of websites into good or bad. We also analyzed the performance of various machine learning algorithms and one statistical method to find the best suited for accurately classifying the websites.

The main contributions of this report are summarized as follows: First, we collected 3 sets of data from 20 categories of Pixel Awards for each year from 2009 to 2011, considering homepages and 1-level web pages for each website. In this way, we have the dataset for 2678 web pages from 255 websites. Second, we computed 9 web page metrics for these web pages using a JAVA based tool. Third, we applied logistic regression and machine learning methods such as MLP, Naïve Bayes, Decision Tree, Bagging, Random Forest, AdaBoost, Random Tree and Decision Table to predict the effect of web page metrics on the classification of web pages into good or bad classes. Although, this research analysis is conducted on 3 datasets, this study can be repeated on different data sets to generalize our findings. Our main results are summarized as follows:

1. The most significant metrics in dataset of year 2009 are Link Count & Page Title Length. Link Count, List Count, Word Count & Page Size were significant metrics in year 2010 and Link Count, List Count, Page title Length, Word Count and Body Text Words in year 2011. This signifies that year by year, the various attributes were included as important metrics to be considered for website development.

2. Naïve Bayes Model outperformed the other models although all the models predicted good area under ROC Analysis. The AUC of Naïve bayes method is within 0.843 - 0.923 in all datasets.

6.1 Application of the Work

In this work, we are able to establish a quantitative relationship between web page metrics and quality of website. Pixel Awards are the website award that honors the websites showing great design and development. However, the judging criteria for such websites are so broad, that it is not possible for the designers to understand the criteria and improve their websites. Thus, this work can be applied to provide the designers with the important metrics that must be considered for the website design as well as the model that is useful for assessing the quality of the website. These metrics will help the designers for ensuring the quality of their website. One can easily assess the quality by simply computing the values of the metrics and applying the Naive Bayes model which is found to be most effective in our work. The websites which are found to be bad will need extra attention and can be improved further. The researchers and website designers can apply Naïve Bayes model for the quality assessment of the website.

6.2 Future Work

This study confirms that construction of models using the machine learning methods is feasible, and useful in predicting the goodness of the websites. More similar type of studies must be carried out with the different data sets and using the different metrics as well to give generalized results. We plan to carry our research for all the levels of web pages in the website and to propose some more web page metrics that are more effective for the websites.

REFERENCES

[1] I. Alsmadi, A. T. Al-Taani and N. A. Zaid, “Web Structural Metrics Evaluation,” In *Developments in E-systems Engineering (DESE)*. 2010.

[2] L. Breiman, “Bagging Predictors,” In *Machine Learning Journal* 26(2), 123-140, 1996.

[3] L. Breiman, “Random Forests,” In *Machine Learning Journal* 45 5- 32, 2001.

[4] E. H. Chi, P. Pirolli and J. Pitkow, “The scent of a site: A system for analyzing and predicting information scent, usage, and usability of a web site,” In *Proceedings of ACM CHI 00 Conference on Conference on Human Factors in Computing Systems,* 2000.

[5] F. De Souza and N. Bevan, “The use of guidelines in menu interface design: Evaluation of a draft standard,” In *Proceedings of IFIP INTERACT’90: Human-Computer Interaction*, Detailed Design: Menus, pages 435–440, 1990.

[6] M. C. Drott, “Using web server logs to improve site design,” In *ACM 16th International Conference on Systems Documentation*, pages 43–50, 1998.

[7] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” In *Journal of computer and system sciences* 55, 119_139, 1997.

[8] R. Fuller and J. J. De Graaff, “Measuring user motivation from server log files,” In *Proceedings of the Human Factors and the Web 2 Conference*, 1996.

[9] H. He, "Decision Tree Learning," Applied Machine Learning. Resource Document. http://www.ece.stevens-tech.edu/~hhe/cpe695f09/lecturenotes/Lecture3.pdf, 2009. Accessed 30 January 2012.

[10] M. Y. Ivory, R. Sinha, and M. A. Hearst, "Preliminary findings on quantitative measures for distinguishing highly rated information-centric web pages," In *Proceedings of the 6th Conference on Human Factors and the Web*, 2000.

[11] M. Y. Ivory, R. Sinha, and M. A. Hearst, "Empirically Validated Web Page Design Metrics," In *Proceedings of the SIGCHI conference on Human factors in computing systems*, p.53-60, Seattle, Washington, United States, 2001.

[12] M. O. Leavitt, and B. Shneiderman, "*Research Based Web Design and Usability Guidelines*," U.S. Department of Health and Human Services,Washington, 2006.

[13] P. Li and S. Yamada, "Automated Web Site Evaluation – An Approach Based on Ranking SVM," In *International Joint Conferences on Web Intelligence and Intelligent Agent Technologies,* IEEE/WIC/ACM, 2009.

[14] P. Li and S. Yamada, "Extraction of Web Site Evaluation Criteria and Automatic Evaluation," In *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol.14 No.4, 2010.

[15] J. Nielsen, "*Designing Web Usability: The Practice of Simplicity*," New Riders Publishing, Indianapolis, IN, 2000.

[16] Pixel Awards | Web Awards Competition (2006). Available from http://www.pixelawards.com.

[17] J. Scholtz, S. Laskowski and L. Downey, “Developing usability tools and techniques for designing and testing web sites,” In *Proceedings of the 4th Conference on Human Factors & the Web,* 1998.

[18] Y. Singh, R. Malhotra, and P. Gupta, “Empirical Validation of Web Metrics for Improving the Quality of Web Page,” In *International Journal of Advanced Computer Science and Applications (IJASCA),* Vol. 2, No. 5, 2011.

[19] M. Stone, “Cross-validatory choice and assessment of statistical predictions,” In *Journal of the Royal Statistical Society, Series B (Methodological),* 36, 111–147, 1974.

[20] H. Thimbleby, “Gentler: A tool for systematic web authoring,” In *International Journal of Human-Computer Studies*, 47, 139-168, 1997.

[21] G. Velayathan and S. Yamada, “Behavior-Based Web Page Evaluation,” In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT Workshops)*,2006.

[22] Weka 3: Data Mining Software in Java. Available from http://www.cs.waikato.ac.nz/ml/weka/. Accessed 5 December 2011.

[23] P. Warren, C. Gaskell, and C. Boldyreff, "Preparing the Ground for Website Metrics Research," *Proceedings of the 3rd International Workshop on Web Site Evolution (WSE 2001),* Florence, Italy, IEEE Computer Press, 2001.

[24] M. Zorman, V. Podgorelec, P. Kokol, and S. H. Babic, “Using machine learning techniques for automatic evaluation of Websites,” *Proceedings of the Third International Conference on Computational Intelligence and Multimedia Applications*

ICCIMA '99, pp. 169-173, New Delhi, India, IEEE Computer Society Press, September 1999.

[25] K. M. Khan, "Assessing Quality of Web Based System," *IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, Page(s): 763 – 769, 2008.

[26] J. A. Anderson. An Introduction to Neural Networks. *Prentice Hall*, 2003.

[27] I. H. Witten, E. Frank, and M.A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques," *Morgan Kaufmann, San Francisco,* 3 edition, 2011.

[28] Y. Freund, and R.E. Schapire, "A short introduction to boosting," In *Journal of Japanese Society for Artificial Intelligence,* 14(5):771-780, September 1999. Appeared in Japanese, translation by Naoki Abe.

[29] K. P. Murphy, "Naive Bayes Classifiers," *Technical Report,* October 2006.

[30] A. Liaw and M. Wiener. Classification and regression by randomForest. In *R News*, 2/3:18-22, December 2002. URL http://www.r-project.org/doc/Rnews/Rnews_2002-3.pdf

[31] M.Hall, "Correlation-based feature selection for discrete and numeric class machine learning," In *Proceedings of the 17th International Conference on Machine Learning*, pp.359-366.

[32] M.A. Hall, "Correlation-based Feature Subset Selection for Machine Learning," PhD thesis, Department of Computer Science, University of Waikato, Waikato, N.Z., 1999.