ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this project work. I would like to take this opportunity to thank them all.

First and foremost I would like to thank **Dr. B. S. Satyanarayana**, Principal, R.V.C.E, Bengaluru, for his moral support towards completing my project work.

I would like to thank **Dr.G.Shobha**, Head of Department, Computer Science & Engineering, R.V.C.E, Bengaluru, for her valuable suggestions and expert advice.

I deeply express my sincere gratitude to my guide Mrs. **Shanta Rangaswamy,** Asst. Prof**,** Department of CSE, R.V.C.E, Bengaluru, for her able guidance, regular source of encouragement and assistance throughout this project

I thank my Parents, and all the Faculty members of Department of Computer Science & Engineering for their constant support and encouragement.

Last, but not the least, I would like to thank my peers and friends who provided me with valuable suggestions to improve my project.

Abstract

Innovation in the field of green technology by analyzing and developing efficient algorithms to reduce CPU utilization is the need of the hour. Managing the vast amount of online information is an important step towards this need. In the field of machine learning, binary classification algorithms are one such technique to process data. This data can have multiple sources and representations which influence the choice of classification technique. Data from web pages is unstructured in nature, and therefore requires many different processes and stages in classification. The aim of this project is two-fold: first, to compare important binary classification algorithms' performance in classifying samples of a large dataset from dynamic sources in the domain of online course websites, to identify valid online courses and secondly to implement the most precise algorithm in building an offline repository of all these courses for a user to access through an interactive front end.

The project involves multiple techniques such as crawling, URL filtering, metadata analysis, text extraction and processing while incorporating novel features such as social tagging from social network websites such as Delicious. A sample dataset covering all corner cases and representative examples is used for training a model classifier, which is then trained and tested using various methods like bootstrapping and cross validation. Finally the efficiency of the model is compared across five algorithms : Naive Bayes, K Nearest Neighbours, Decision Trees, Logistic Regression and Random Forest on datasets of increasing size and the most efficient one is used for classifying all the data collected from crawling. The relevant courses are put into an offline database made available to the users by an interactive and fresh user interface.

The parameter used to evaluate the algorithms was the accuracy of predictions on the test set, namely precision. It was observed that Random Forest algorithm outperforms Naive Bayes, K Nearest Neighbours, Decision Tree and Logistic Regression on this parameter. The accuracy observed upon evaluation of the test set for this classifier is 92.7%. Over 30,000 URLs were crawled and pre-processed using this classifier to create a repository of over 10,000 valid online courses.

Table of Contents

Acknowledgement **i**

Abstract **ii**

Table of Contents **iii**

List of abbreviations **vii**

List of Tables **viii**

List of Figures **x**

Chapter 1

1 Introduction 1

1.1 State of art developments 2
1.2 Motivation 13
1.3 Problem Statement 13
1.4 Objective 14
1.5 Scope 14
1.6 Methodology 14
1.7 Organization of the Report 15
1.8 Summary 16

Chapter 2

2 Software Requirements Specification of classification of unstructured data from online course web pages 17

2.1 Overall Description 17

2.1.1 Product Perspective 17
2.1.2 Product Functions 18
2.1.3 User Characteristics 18
2.1.4 Constraints 18
2.1.5 Assumptions and Dependencies 19

2.2 Specific Requirements 19

2.2.1 Functionality Requirements 19
2.2.2 Performance Requirements 19
2.2.3 Supportability 20
2.2.4 Software Requirements 20
2.2.5 Hardware Requirements 20
2.2.6 Design Constraints 21

2.2.7 Interfaces 21

2.3 Other Non-functional Requirements 22

2.3.1 Performance Requirements 22

2.3.2 Safety and Security Requirements 22

2.3.3 Software Quality Attributes 23

2.4 Summary 24

Chapter 3

3 High level design of classification of unstructured data from online course web pages 25

3.1 Design constraints 25

3.1.1 General constraints 25

3.1.2 Development methods 26

3.2. Architectural strategies 26

3.2.1 Programming language 26

3.2.2 Future plans 27

3.2.3 Error detection and recovery 27

3.2.4 Data Storage management 27

3.2.5 Communication mechanism 27

3.3 System architecture 28

3.4 Data Flow Diagrams 28

3.4.1 DFD-level 0 28

3.4.2 DFD-level 1 29

3.4.3 DFD-level 2 30

3.5 Summary 30

Chapter 4

4 Detail design of classification of unstructured data from online course web pages 32

4.1 Structure chart 32

4.2 Function description of modules 33

4.2.1 Crawler module 34

4.2.2 Featurizer module 35

4.2.3 Classifier module 37

4.2.4 Database module 38

4.2.5 User interface module 40

4.3 Algorithms 41

4.3.1 C4.5 algorithm 41

4.3.2 Logistic Regression algorithm 42

4.3.3 Naïve bayes algorithm 44

4.3.4 K nearest neighbour algorithm 45

4.3.5 Random Forest algorithm 45
4.3 Summary 46

Chapter 5

5 Implementation of classification of unstructured data from online course web pages 47

5.1 Programming language selection 47
5.2 Platform selection 47
5.3 Code conventions 47
5.3.1 Naming conventions 48
5.3.2 File organizations 48
5.3.3 Properties declaration 49
5.3.4 Class declarations 49
5.3.5 Comments 49
5.4 Difficulties Encountered and Strategies Used to Tackle 50
5.4.1 Crawling and Classification Logic 50
5.4.2 Storage and Display Logic 51
5.5 Summary 51

Chapter 6

6 Software testing of classification of unstructured data from online course web pages 52

6.1 Testing environment 52
6.2 Unit Testing of Main Modules 53
6.2.1 Unit Testing of Crawler Module 53
6.2.2 Unit Testing of Featurizer Module 54
6.2.3 Unit Testing of Classifier Module 55
6.2.4 Unit Testing of Database Module 56
6.2.5 Unit Testing of User Interface Module 57
6.3 Integration Testing Of Modules 57
6.4 System testing 59
6.5 Summary 60

Chapter 7

7 Experimental results and analysis of classification of unstructured data from online course web pages 61

7.1 Evaluation metrics 61
7.2 Experimental dataset 62
7.3 Performance Analysis 62
7.3.1 C4.5 Algorithm Evaluation 63

7.3.2 Logistic Regression Algorithm Evaluation 65
7.3.3 Naive Bayes Algorithm Evaluation 68
7.3.4 K Nearest Neighbours Algorithm Evaluation 70
7.3.5 Random Forest Evaluation 73
7.4 Inference from the result 75
7.5 Summary 77

Chapter 8

8 Conclusion 78

8.1 Limitation 78
8.2 Future enhancements 79
8.3 Summary 79
References 80

Appendices 85

Appendix A 85
Appendix B 96

List of abbreviations

AI	:	Artificial Intelligence
ANFIS	:	Adaptive Neuro Fuzzy Inference System
FoCUS	:	Crawler Under Supervision
DOM	:	Document Object Module
NE	:	Named Entities
HMM	:	Hidden Markov Model
ME	:	Maximum Entropy
TBL	:	Transformation Based error-driven Learning
SVM	:	Support Vector Machine
JDBC	:	Java Database Connection
UDP	:	User Datagram Protocol
TCP	:	Transmission Control Protocol
OS	:	Operating System
HCL	:	Hardware Compatibility List
CPU	:	Central Processing Unit
MIPS	:	Million Instructions Per Second
GUI	:	Graphical User Interface
CSV	:	Common Separated Value
DBF	:	Database File
SQL	:	Structured Query Language
SDLC	:	Software Development Life Cycle
DFD	:	Data Flow Diagram
CLD	:	Context Level Diagram
API	:	Application Programming Interface
REST	:	Representational state transfer

List of Tables

Table No.	Title	Page No.
Table 6.1:	Unit Test Case 1: Checks if Basic crawler works	53
Table 6.2:	Unit Test Case 2: Checks if multiple crawler works	54
Table 6.3:	Unit Test Case 3: Check if featurizer works	55
Table 6.4:	Unit Test Case 4: Check if featurizer works	55
Table 6.5:	Unit Test Case 5: Check if classifier works	56
Table 6.6:	Unit Test Case 6: Checks if Database module works	56
Table 6.7:	Unit Test Case 7: Checks if User Interface module works	57
Table 6.8:	Integrated test case 1: Test case to see if search option in the user interface works	58
Table 6.9:	Integrated test case 2: Test case to see if update option in the user interface works	58
Table 6.10:	Integrated test case 3: Test case to see if bookmark option in the user interface works	59
Table 6.11:	System Testing	60
Table 7.1:	Detailed accuracy by class on training set for C4.5 algorithm	63
Table 7.2:	Confusion matrix of training set for C4.5 algorithm	63
Table 7.3:	Detailed accuracy by class on test set for C4.5 algorithm	64
Table 7.4:	Confusion matrix of test set for C4.5 algorithm	64
Table 7.5:	Detailed accuracy by class on training set for Logistic Regression algorithm	65
Table 7.6:	Confusion matrix of training set for C4.5 for Logistic Regression algorithm	66
Table 7.7:	Detailed accuracy by class on test set for Logistic Regression algorithm	66
Table 7.8:	Confusion matrix of training test set for Logistic Regression algorithm	66

Table 7.9: Detailed accuracy by class on training set for Naïve Bayes algorithm 68

Table 7.10: Confusion matrix of training set for Naïve Bayes algorithm 68

Table 7.11: Detailed accuracy by class on test set for Naïve Bayes algorithm 68

Table 7.12: Confusion matrix of test set for Naïve Bayes algorithm 69

Table 7.13: Detailed accuracy by class on training set for K-Nearest Neighbor algorithm 70

Table 7.14: Confusion matrix of training set for k-Nearest Neighbor algorithm 71

Table 7.15: Detailed accuracy by class on test set for K-Nearest Neighbor algorithm 71

Table 7.16: Confusion matrix of test set for k-Nearest Neighbor algorithm 71

Table 7.17: Detailed accuracy by class on training set Random Forest algorithm 73

Table 7.18: Confusion matrix of training set for Random Forest algorithm 73

Table 7.19: Detailed accuracy by class on test set for Random Forest algorithm 73

Table 7.20: Confusion matrix of test set for Random Forest algorithm 74

Table 7.21: Accuracies of the algorithms 76

List of Figures

Figure No.	Title	Page No.
Figure 1.1:	General architecture of implementation of machine learning algorithms for efficient classification of unstructured data from online course web pages	2
Figure 3.1:	DFD-level 0 for online course webpage classification system	29
Figure 3.2:	DFD-level 1 for online course webpage classification system	29
Figure 3.3:	DFD-level 2 for online course webpage classification system	30
Figure 4.2:	Structure chart of implementation of machine learning algorithms for efficient classification of unstructured data from online course web pages	33
Figure 4.3:	Flow chart for crawler	35
Figure 4.4:	Flow chart for featurizer	36
Figure 4.5:	Flow chart for classifier	38
Figure 4.6:	Flowchart for database module	39
Figure 4.7:	Flow chart for user interface module	40
Figure 4.8:	The logistic function graph	44
Figure 7.1:	ROC curve of C4.5 classification algorithm for the class value “yes”	64
Figure 7.2:	ROC curve of C4.5 classification algorithm for the class value “no”	65
Figure 7.3:	ROC curve of Logistic Regression classification algorithm for the class value “no”	67
Figure 7.4:	ROC curve of Logistic Regression classification algorithm for the class value “yes”	67
Figure 7.5:	ROC curve of Naive Bayes classification algorithm for the class value ‘no’	69
Figure 7.6:	ROC curve of Naive Bayes classification algorithm for the class value ‘yes’	70
Figure 7.7:	ROC curve of K Nearest classification algorithm for the class value no’	72
Figure 7.8:	ROC curve of K Nearest classification algorithm for the class value ‘no’	72
Figure 7.9:	ROC curve of Random Forest classification algorithm for the class value ‘no’	74
Figure 7.10:	ROC curve of Random Forest classification algorithm for the class value ‘yes’	75