Abstract

Author profiling is the task of identifying different personality traits of author by analysing their written text. Different personality traits are author's gender, age, native language, native city, qualification, occupation etc. Author profiling has different applications such as it could be used in forensics to find the suspect, in marketing, for fake profile identification etc. We aim to predict the author's age and gender from the multilingual SMS. In this paper we have classify the author gender into male or female categories. Moreover we have classify the author age into three different groups: (i) 15-19, (ii) 20-24, (iii) above 25. Several machine learning classification models like Support Vector Machine, Random Forest, Decision Tree, Naïve Bayes and Linear Regression are used for experiment. Experimental Results show that Support Vector Machine provide better accuracy when used with Count Vectorizer and TFIDF (Term Frequency Inverse Document Frequncy).

Acknowledgement

A research project like this is never the work of any one alone. The contributions of many different people, in their different ways, have made this possible. We would like to extend our appreciation especially to the following.

It is our pleasure to take this opportunity to thank all those who helped us directly or indirectly in our research work. Not everything that we have received can be acknowledged with few words.

We are thankful to our guide **Dr. Hiren Joshi** & **Mr. Hardik Joshi** for their valuable help during the work of this thesis. Their suggestions were always there whenever we needed and their guidance helped us in all the time of research and writing of this thesis.

We articulate our deep sense of respect and gratitude to **Dr. Savita Gandhi** (Director of Department of Computer Science, Gujarat University). We are also thankful to all the faculties of **Department of Computer Science, Rollwala Computer Centre**

Our deepest thanks to all staff of lab in Department of Computer Science, Gujarat University for providing necessary infrastructure for the research.

Shifa Khan

Aatithya Hitkar

Table of Contents

List of Figures 7
1. Introduction 10
1.1 Information Retrieval 10
1.2 History Information Retrieval 10
1.3 IR Model Types 11
1.3.1 First dimension: mathematical basis 11
1.3.2 Second dimension: properties of the model 11
1.4 Timeline 11
1.5 Major conferences/forums 15
1.6 Awards in the field 15
1.7 Top 5 Leading IR Research Groups 15
1.8 What is Multilingual? 16
1.9 What is Profiling ? 16
1.10 What is Author Profiling ? 16
1.11 What is Multilingual Author Profiling on SMS? 16
1.12 Research Problem 16
2. Overview of Literature Review 18
2.1 Author Gender Identification from Text 18
2.2 Identifying Gender from SMS Text Messages 19
2.3 Use of Language and Author Profiling 20
2.4 Author Profiling: Predicting Age and Gender from Blogs 21
2.5 Inferring Gender of Movie Reviewers 22
2.6 Author Profiling in the wild 23
2.7 Multilingual author profiling on Facebook 24
2.8 Author Profiling for Blogs 25
2.9 Author Profiling Using Word Embedding Averages 26
2.10 Multilingual SMS-based Author Profiling 27
2.11 Language and Gender Author Cohort Analysis of E-mail for Computer Forensics 28
2.12 Automatically Profiling the Author of an Anonymous Text 29
2.13 Gender Prediction in English 30
2.14 Effects of Age and Gender on Blogging 31
2.15 Author Profile Prediction using Pivoted Unique Term Normalization 32
2.16 Identifying the Author of Email 33
2.17 N-gram approach for gender prediction 34
2.18 Effects of Age and Gender on Blogging 35

3. Language Processing with Python 37
3.1 What is NLP? 37
3.2 History of NLP 37
3.3 How NLP Works ? 38
3.4 Components of NLP 39
3.5 NLP and Writing Systems 41
3.6 How to Implement NLP? 42
3.7 Advantages of NLP 42
3.8 Disadvantages of NLP 42
3.9 What is NLTK? 43
3.10 Installing NLTK in Linux 43
3.11 Installing NLTK through Anaconda 44
3.12 How to download all packages of NLTK? 45
3.13 Tokenization 45
3.14 NLTK stopwords 46
3.15 NLTK Stemming 47
3.16 TFIDF – Term Frequency Inverse Document Frequency 48
3.17 What are N-Gram? 50
4. Machine Learning Approach 52
4.1 What is Machine Learning? 52
4.2 How Machine Learning Works? 52
4.3 Types of Machine Learning 53
4.4 What is Supervised Learning? 53
4.5 What is Unsupervised Learning? 53
4.6 What is Reinforcement Learning? 53
4.7 Python Libraries for Machine Learning 54
4.7.1 Numpy 55
4.7.2 Scikit-learn 56
4.7.3 TensorFlow 57
4.7.4 Pandas 57
4.7.5 Matplotlib 58
4.8 Machine Learning Classification 59
4.8.1 Naïve Bayes 59
4.8.2 Support Vector Machine 60
4.8.3 Decision Trees 66
4.9 Random Forest 69

4.9.1 Logistic Regression 71
5. Proposed Model 74
5.1 Research Problem 74
5.2 Applications 75
5.3 Dataset Classification 75
5.4 Sample Data : Male 76
5.5 Sample Data : Female 76
5.6 Tags Identification 76
5.7 Different approaches 78
5.8 Work Flow 80
5.9 Tokenization 82
5.10 Pre-processing of data 82
5.11 Pre-Processed data 83
5.12 Tokenization after pre-processing 84
5.13 Tokenization after pre-processing 84
5.14 Term Frequency 85
5.15 TFIDF – Dataset 85
6. Experiments and results 87
6.1 Naïve Bayes 87
6.2 Support Vector Machine 88
6.3 Decision Tree 89
6.4 Random Forest 90
6.5 Experiment Analysis 91
6.6 Conclusion 92
6.7 Future Work 92
7. Bibliography and Citations 93

List of Figures

Figure 3-1 Word Vector Example 37

Figure 3-2 Component of NLP 39

Figure 3-3 Installing Anaconda 43

Figure 3-4 Installing Steps of NLTK 43

Figure 3-5 TF-IDF Calculation 48

Figure 4-1 Working of Machine Learning 51

Figure 4-2 Classification algorithm in ML 53

Figure 4-3 Matplotlib Vector 57

Figure 4-4 Data Label 60

Figure 4-5 Multiple Hyperplane 60

Figure 4-6 Non Linear Data 61

Figure 4-7 Multi-Dimensional in SVM 61

Figure 4-8 Best Hyperplane in SVM 62

Figure 4-9 Tree Data Splitting 63

Figure 4-10 Tree Pruning 66

Figure 4-11 Random Forest Tree 67

Figure 4-12 Linear Regression Accuracy 68

Figure 5-1 Dataset Classification 71

Figure 5-2 Dataset Graph 74

Figure 5-3 Sample data of male 74

Figure 5-4 Sample data of female 75

Figure 5-5 Work Flow 75

Figure 5-6 Raw Data 79

Figure 5-7 Tokenization 80

Figure 5-8 Pre-processing of data 81

Figure 5-9 Pre-processed data 81

Figure 5-10 Tokenization after pre-processing 82

Figure 5-11 Term Frequency 82

Figure 5-12 TF-IDF of dataset 83

Figure 6-1 Naïve Bayes result 86

Figure 6-2 SVM result 87

Figure 6-3 Decision Tree result 88

Figure 6-4 Random Forest result 89

Figure 6-5 Experiment analysis 90

Chapter 1

Introduction

1. Introduction

1.1 Information Retrieval

Information retrieval (IR) is the activity of obtaining information system resources relevant to an information need from a collection of information resources. Searches can be based on full-text or other content-based indexing. Information retrieval is the science of searching for information in a document, searching for documents themselves, and also searching for metadata that describe data, and for databases of texts, images or sounds.

Automated information retrieval systems are used to reduce what has been called information overload. An IR system is a software that provide access to books, journals and other documents, stores them and manages the document. Web search engines are the most visible IR applications.[1]

1.2 History Information Retrieval

The idea of using computers to search for relevant pieces of information was popularized in the article As We May Think by Vannevar Bush in 1945. It would appear that Bush was inspired by patents for a 'statistical machine' - filed by Emanuel Goldberg in the 1920s and '30s - that searched for documents stored on film. The first description of a computer searching for information was described by Holmstrom in 1948, detailing an early mention of the Univac computer. Automated information retrieval systems were introduced in the 1950s: one even featured in the 1957 romantic comedy, Desk Set. In the 1960s, the first large information retrieval research group was formed by Gerard Salton at Cornell. By the 1970s several different retrieval techniques had been shown to perform well on small text corpora such as the Cranfield collection (several thousand documents). Large-scale retrieval systems, such as the Lockheed Dialog system, came into use early in the 1970s. [1]

1.3 IR Model Types

1.3.1 First dimension: mathematical basis

- Standard Boolean model
- Extended Boolean model
- Fuzzy retrieval

1.3.2 Second dimension: properties of the model

Models without term-interdependencies treat different terms/words as independent. This fact is usually represented in vector space models by the orthogonality assumption of term vectors or in probabilistic models by an independency assumption for term variables.

Models with immanent term interdependencies allow a representation of interdependencies between terms. However the degree of the interdependency between two terms is defined by the model itself. It is usually directly or indirectly derived (e.g. by dimensional reduction) from the co-occurrence of those terms in the whole set of documents.

Models with transcendent term interdependencies allow a representation of interdependencies between terms, but they do not allege how the interdependency between two terms is defined. They rely an external source for the degree of interdependency between two terms. [1]

1.4 Timeline

Before the 1900s

1801: Joseph Marie Jacquard invents the Jacquard loom, the first machine to use punched cards to control a sequence of operations.

1880s: Herman Hollerith invents an electro-mechanical data tabulator using punch cards as a machine readable medium.

1890 Hollerith cards, keypunches and tabulators used to process the 1890 US Census data.

1920s-1930s

Emanuel Goldberg submits patents for his "Statistical Machine" a document search engine that used photoelectric cells and pattern recognition to search the metadata on rolls of microfilmed documents.

1940s–1950s

Late 1940s: The US military confronted problems of indexing and retrieval of wartime scientific research documents captured from Germans.

1945: Vannevar Bush's As We May Think appeared in Atlantic Monthly.

1947: Hans Peter Luhn (research engineer at IBM since 1941) began work on a mechanized punch card-based system for searching chemical compounds.

1950s: Growing concern in the US for a "science gap" with the USSR motivated, encouraged funding and provided a backdrop for mechanized literature searching systems (Allen Kent et al.) and the invention of citation indexing (Eugene Garfield).

1950: The term "information retrieval" was coined by Calvin Mooers.

1951: Philip Bagley conducted the earliest experiment in computerized document retrieval in a master thesis at MIT.

1955: Allen Kent joined Case Western Reserve University, and eventually became associate director of the Center for Documentation and Communications Research. That same year, Kent and colleagues published a paper in American Documentation describing the precision and recall measures as well as detailing a proposed "framework" for evaluating an IR system which included statistical sampling methods for determining the number of relevant documents not retrieved.

1958: International Conference on Scientific Information Washington DC included consideration of IR systems as a solution to problems identified. See: Proceedings of the International Conference on Scientific Information, 1958 (National Academy of Sciences, Washington, DC, 1959)

1959: Hans Peter Luhn published "Auto-encoding of documents for information retrieval." [1]

1960s

Early 1960s: Gerard Salton began work on IR at Harvard, later moved to Cornell.

1960: Melvin Earl Maron and John Lary Kuhns published "On relevance, probabilistic indexing, and information retrieval" in the Journal of the

1962:

Cyril W. Cleverdon published early findings of the Cranfield studies, developing a model for IR system evaluation. See: Cyril W. Cleverdon, "Report on the Testing and Analysis of an Investigation into the Comparative Efficiency of Indexing Systems". Cranfield Collection of Aeronautics, Cranfield, England, 1962.

Kent published Information Analysis and Retrieval.

1963:

Weinberg report "Science, Government and Information" gave a full articulation of the idea of a "crisis of scientific information." The report was named after Dr. Alvin Weinberg.

Joseph Becker and Robert M. Hayes published text on information retrieval. Becker, Joseph; Hayes, Robert Mayo. Information storage and retrieval: tools, elements, theories. New York, Wiley (1963).

1964:

Karen Spärck Jones finished her thesis at Cambridge, Synonymy and Semantic Classification, and continued work on computational linguistics as it applies to IR.

The National Bureau of Standards sponsored a symposium titled "Statistical Association Methods for Mechanized Documentation." Several highly significant papers, including G. Salton's first published reference (we believe) to the SMART system. [1]

Mid-1960s

National Library of Medicine developed MEDLARS Medical Literature Analysis and Retrieval System, the first major machine-readable database and batch-retrieval system.

Project Intrex at MIT.

1965: J. C. R. Licklider published Libraries of the Future.

1966: Don Swanson was involved in studies at University of Chicago on Requirements for Future Catalogs.

Late 1960s: F. Wilfrid Lancaster completed evaluation studies of the MEDLARS system and published the first edition of his text on information retrieval.

1968:

Gerard Salton published Automatic Information Organization and Retrieval.

John W. Sammon, Jr.'s RADC Tech report "Some Mathematics of Information Storage and Retrieval..." outlined the vector model.

1969: Sammon's "A nonlinear mapping for data structure analysis" (IEEE Transactions on Computers) was the first proposal for visualization interface to an IR system. [1]

Early 1970s

First online systems—NLM's AIM-TWX, MEDLINE; Lockheed's Dialog; SDC's ORBIT.

Theodor Nelson promoting concept of hypertext, published Computer Lib/Dream Machines.

1971: Nicholas Jardine and Cornelis J. van Rijsbergen published "The use of hierarchic clustering in information retrieval", which articulated the "cluster hypothesis."

1975: Three highly influential publications by Salton fully articulated his vector processing framework and term discrimination model:

A Theory of Indexing (Society for Industrial and Applied Mathematics)

A Theory of Term Importance in Automatic Text Analysis (JASIS v. 26)

A Vector Space Model for Automatic Indexing (CACM 18:11)

1978: The First ACM SIGIR conference.

1979: C. J. van Rijsbergen published Information Retrieval (Butterworths). Heavy emphasis on probabilistic models.

1979: Tamas Doszkocs implemented the CITE natural language user interface for MEDLINE at the National Library of Medicine. The CITE system supported free form query input, ranked output and relevance feedback. [1]

1980s

1980: First international ACM SIGIR conference, joint with British Computer Society IR group in Cambridge.

1982: Nicholas J. Belkin, Robert N. Oddy, and Helen M. Brooks proposed the ASK (Anomalous State of Knowledge) viewpoint for information retrieval. This was an important concept, though their automated analysis tool proved ultimately disappointing.

1983: Salton (and Michael J. McGill) published Introduction to Modern Information Retrieval (McGraw-Hill), with heavy emphasis on vector models.

1985: David Blair and Bill Maron publish: An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System. [1]

Mid-1980s

Efforts to develop end-user versions of commercial IR systems.

1985–1993: Key papers on and experimental systems for visualization interfaces.

Work by Donald B. Crouch, Robert R. Korfhage, Matthew Chalmers, Anselm Spoerri and others.

1989: First World Wide Web proposals by Tim Berners-Lee at CERN.

1990s

1992: First TREC conference.

1997: Publication of Korfhage's Information Storage and Retrieval with emphasis on visualization and multi-reference point systems.

1999: Publication of Ricardo Baeza-Yates and Berthier Ribeiro-Neto's Modern Information Retrieval by Addison Wesley, the first book that attempts to cover all IR. [1]

Late 1990s

Web search engines implementation of many features formerly found only in experimental IR systems. Search engines become the most common and maybe best instantiation of IR models. [1]

1.5 Major conferences/forums

SIGIR: Conference on Research and Development in Information Retrieval

ECIR: European Conference on Information Retrieval

CIKM: Conference on Information and Knowledge Management

WWW: International World Wide Web Conference

WSDM: Conference on Web Search and Data Mining

ICTIR: International Conference on Theory of Information Retrieval

TREC: Text REtrieval Conference

CLEF: Conference and Lab of the Evaluation Forum

NTCIR: NII Testbeds for Information Access Research.

FIRE: Forum for Information Retrieval and Evaluation

1.6 Awards in the field

Tony Kent Strix award

Gerard Salton Award

1.7 Top 5 Leading IR Research Groups

- Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts Amherst
- Information Retrieval Group at the University of Glasgow
- Information and Language Processing Systems (ILPS) at the University of Amsterdam
- Information Retrieval Group (THUIR) at Tsinghua University
- Information Storage, Analysis and Retrieval Group (ISAR) at RMIT University. [1]

1.8 What is Multilingual?

Multilingualism is the use of more than one language, either by an individual speaker or by a community of speakers. It is believed that multilingual speakers outnumber monolingual speakers in the world's population. More than half of all Europeans claim to speak at least one language other than their mother tongue.

Using more than one language in communication.

Co-presence of two or more languages in a text, speech or society.

Example: Hindi – English, Hindi – Gujarati, Gujarati - English.

1.9 What is Profiling ?

The recording and analysis of a person's psychological and behavioural characteristics, so as to assess or predict their capabilities in a certain sphere or to assist in identifying categories of people.

1.10 What is Author Profiling ?

A text Classification technique that is used to predict the profiling characteristics of the authors like gender, age, native language and educational background by analysing their text."

It is based on stylistic and content-based features.

1.11 What is Multilingual Author Profiling on SMS?

Finding demographic features like age, gender, native language of an author from the written text. We will be using Roman Hindi and English as our Multilingual Data. Roman Hindi (written using English Alphabets) is used in daily communication by a large number of people. It is used in comments, Tweets, Blogs, SMS -messages etc.

Majority of the research on Author Profiling is done for English, Spanish, Italian and Dutch Language. [2]

1.12 Research Problem

We aim to find the Author's age and Gender from the messages which are written in Roman Hindi and English.

Gender Identification:

To classify the multilingual author to Male or Female

Age Identification:

To classify the multilingual author profile into one of the three categories. 15-19, 20-24, 25-XX.

Chapter 2

Overview of Literature Review

2. Overview of Literature Review

2.1 Author Gender Identification from Text

Authors	Journal	Year	Summary
Na Cheng, R.Chandramauli, K.P subbalakshmi.	Digital Investigation, 2011 – Elsevier	2011	**Methods**:- Support Vector Machine, Logistic Regression and AdaBoost Decision Tree **Accuracy**:- Gender - 83%

Proposed System:

Developed short length author gender identification.

They has proposed psycho-linguistic with stylometric features to solve the identification problem.

Three Machine Learning Algorithms like Support Vector Machine, Logistic Regression and AdaBoost Decision Tree were designed for gender identification

Observation:

Support Vector Machine performs better than Logistic Regression and AdaBoost Decision Tree.

This design gives 82% accuracy when the text is short. If the text is large then it does not provide the better accuracy.

2.2 Identifying Gender from SMS Text Messages

Authors	Journal	Year	Summary
Shannon Silessi, Cihan Varol	15th IEEE Conference on Machine Learning and Application	2012	**Methods**:- Support Vector Machine, Logistic Regression. **Accuracy**:- Gender - 72.39%

Proposed System:

Proposed a method to classify the gender of the author from the sms text messages.

Three Machine Learning Algorithms like Support Vector Machine, Logistic Regression and AdaBoost Decision Tree were used for gender identification.

Observation:

This approach has provided 72.39% accuracy level.

In this approach they had also consider the author name along with the messages for the gender identification.

2.3 Use of Language and Author Profiling

Authors	Journal	Year	Summary
Francisco Rangel, Paolo Rosso	Natural Language Processing and Cognitive 2013	2013	**Methods**:- stylistic features with Support Vector Machine. **Accuracy**:- Gender - 77.87%

Proposed System:

This paper presents the approach to identify age and gender of author based on their use of language.

They had proposed a representation based on stylistic features and obtain result with Support Vector Machine.

They had used the PAN-AP-13 dataset.

Observation:

They had only consider English as a Language.

Men uses more prepositions than women.

Women uses more pronouns, determinants and interjections than men as they are more interested in social relationship.

2.4 Author Profiling: Predicting Age and Gender from Blogs

Authors	Journal	Year	Summary
K Santosh, Romil Bansal, Mihir Shekhar, and Vasudeva Varma	Digital Investigation, 2013 – Elsevier	2013	**Methods:-** Content-based, Style-based, and Topic-based. **Accuracy:-** Gender - 56.53% Age - 64.8%

Proposed System:

They had proposed a machine learning approach to determine unknown author's age and gender from the blogs

The Approach uses three types of features: Content-based, Style-based, and Topic-based.

They had used the blog corpus provided by PAN 2013.

Observation:

They have achieved accuracy of 64.08% for age and 56.53% for gender in English, for Spanish Dataset they had achieved 64.30% for age and 64.73% for gender.

Females tends to write more about wedding styles and male tends to write more about technology, politics and cricket.

Teenagers tend to write more about their friends and mood swings, whereas people of 20's write more about college life and people of 30's write more about marriage, jobs and politics.

2.5 Inferring Gender of Movie Reviewers

Authors	Journal	Year	Summary
Jahna Otterbacher	19th ACM international conference	2017	**Methods:-** Logistic Regression. **Accuracy:-** Gender - 73.7%

Proposed System:

They had determine the gender of the people who gives movie review.

They had used Logistic Regression and explore three types of information: Style, Content and Metadata.

They had used 5000 reviews of 40 movies.

Observation:

They have achieved accuracy of 73.7% for gender classification.

They have considered only English as a Language for Movie Review.

Women describe colors more vividly than Men and women are more likely to use “kind of”, “sort of” words in order to soften an argument.

2.6 Author Profiling in the wild

Authors	Journal	Year	Summary
Lisa Kaati, Elias Lundeqvist, Amendra Shrestha and Maria Svensson	IEEE Explore	2015	**Methods:-** Support Vector Machine **Accuracy:-** Gender - 56%

Proposed System:

They have used machine learning for profiling authors of online textual media.

They had determined the gender and age of an author.

They were interested in understanding how well author profiling can be expected to work when it is trained on one domain and tested on a different domain.

They had used the dataset provided by PAN.

Observation:

When it is used on the same dataset the algorithm provides 73% accuracy but when the testing dataset is changed the accuracy decreases to 56%.

2.7 Multilingual author profiling on Facebook

Authors	Journal	Year	Summary
Mehwish Fatima, Komal Hasan, Saba Anwar, Rao Muhammad Adeel Nawab	Information Processing 2017 - Elsevier	2017	**Methods:-** Support Vector Machine **Accuracy:-** Gender - 56%

Proposed System:

Developed a multilingual (Roman Urdu and English) corpus.

The corpus contains 479 profiles (having both private and public comments/posts of a Facebook user) along with demographic information (age, gender, and personality (type).

Modelling existing author profiling techniques by using content based features (word and character N–grams) and different stylistic based features for age and gender identification on multilingual and translated corpora.

Observation:

Automatically identifies an author's age and gender on proposed multilingual translated corpora, gender with accuracy of 0.875 and age with accuracy of 0.750.

2.8 Author Profiling for Blogs

Authors	Journal	Year	Summary
Dang Duc Pham, Giang Binh Tran, Son Bao Pham	International Conference on Asian Languages Processing ieeeexpolre – 2009	2009	**Methods:-** Random Forest. **Accuracy:-** Gender - 80%

Proposed System:

They have developed a Blog Profiling framework to automatically predict age, gender and occupation of weblogs authors purely based on language use.

They had consider the dataset of 73 bloggers.

They have used the Machine Learning Algorithm Random Forest.

Observation:

The experiments on the blogs provide results with accuracy of around 80%.

Support Vector Machine doesn't work well here as they had taken a small dataset and less features.

2.9 Author Profiling Using Word Embedding Averages

Authors	Journal	Year	Summary
Roy Bayot, Teresa Gonc¸alves	10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA) – 2016	2016	**Methods:-** Support Vector Machine **Accuracy:-** Gender - 45.8% Age - 39.7%

Proposed System:

They have developed a author profiling of tweets in English and Spanish.

They have determined age and gender of author.

They had used the twitter dataset from PAN-2016 and had worked on 430 tweets.

They have used the Support Vector Machine Algorithm.

Observation:

They have achieved accuracy of 39.7% for age classification in English and 53% in Spanish. For Gender Classification they have achieved 45.8% in English and 45.3% in Spanish.

2.10 Multilingual SMS-based Author Profiling

Authors	Journal	Year	Summary
Mehwish Fatima, Saba Anwar, Amna Naveed, Waqas Arshad, Rao Muhammad Adeel Nawab, Muntaha Iqbal and Alia Masood	Natural Language Engineering at Cambridge University Press - 2018	2018	**Methods:-** Support Vector Machine **Accuracy:-** Gender - 95%

Proposed System:

They have developed a author profiling on sms on Urdu and English language.

They had consider the dataset of 810 profiles and had concluded the gender identification from the SMS.

They had applied stylometry-based method and the content-based method for the gender identification

They have used the Support Vector Machine Algorithm.

Observation:

They have obtained the accuracy score of 0.95.

2.11 Language and Gender Author Cohort Analysis of E-mail for Computer Forensics

Authors	Journal	Year	Summary
Malcolm Corney, Alison Anderson, George Mohay, Olivier De Vel	Digital Forensic Research Conference-2017	2017	**Methods:-** Support Vector Machine **Accuracy:-** Gender - 77.8%

Proposed System:

They have developed author profiling on Email in English language.

They had determined gender identification.

Dataset is of 342 Authors with 8820 Emails.

They had applied stylometry-based method and the content-based method for the gender identification

They have used the Support Vector Machine Algorithm.

Observation:

Women's makes more frequent use of emotionally intensive adverbs and adjectives such as "so", "terribly", "awfully", "dreadful" and "quite" and uses more punctuates like assertions, apologies, questions etc.

Men uses strong assertions, aggressive, self-promotion, challenges and humour.

2.12 Automatically Profiling the Author of an Anonymous Text

Authors	Journal	Year	Summary
Shlomo Argamon, Moshe Koppel, James W. Pennebaker	Communication of ACM-2011	2011	**Methods:-** Bayesian Multinomial Regression(BMR) **Accuracy:-** Gender - 42.7%

Proposed System:

They have developed author profiling to identify the gender.

They had applied style-based method and the content-based method for the gender identification.

They have used Bayesian Multinomial Regression(BMR).

They had consider the dataset of 19,320 blog of 258 author in English.

Observation:

Male use more prepositions while Female uses more pronoun.

They have obtained the accuracy of 42.7%.

2.13 Gender Prediction in English

Authors	Journal	Year	Summary
Ankush Khandelwal, Sahil Swami, Syed Sarfaraz Akhtar and Manish Shrivastava	Cornell University Library Arxiv - 2018	2018	**Methods:-** Support Vector Machine **Accuracy:-** Gender - 75.7%

Proposed System:

They have developed author profiling to identify the gender.

They had applied style-based method and the content-based method.

They have used Support Vector Machine (SVM).

They had Considered 1000 tweets as a Dataset.

Observation:

They have obtained the accuracy of 75.7%.

2.14 Effects of Age and Gender on Blogging

Authors	Journal	Year	Summary
Jonathan Schler, Moshe Koppel, Shlomo Argamon, James Pennebaker	Conference of Computational Approaches to Analysing Weblogs – Stanford, California, USA – 2006	2018	**Methods:-** Support Vector Machine **Accuracy:-** Gender - 67.3%

Proposed System:

This paper shows the analysis of 300 million words to indicate differences in writing style and content between male and female blog.

This analysis is used to determine author's age and gender.

They have used 71,000 blogs.

They have used two different kinds of distinguishing features: Style-related and Content-related.

Observation:

Female bloggers use more pronouns and negation words while Male bloggers use more articles and prepositions.

Male bloggers of all ages write more about politics, technology and money.

Female bloggers discuss their personal lives – and use more personal writing style.

2.15 Author Profile Prediction using Pivoted Unique Term Normalization

Authors	Journal	Year	Summary
T. Raghunadha Reddy, B. Vishnu Vardhan and P. Vijayapal Reddy	Indian Journal of Science and Technology - 2016	2016	**Methods:-** Naïve Bayes Multinomial and The logistic Classifier **Accuracy:-** Gender – 68.5%

Proposed System:

This paper proposes a model to predict the profiles of the authors such as gender and age by analyzing their writing styles on hotel reviews dataset.

In this paper, the pivoted unique term normalization measure is used to calculate the weight of the terms specific to each profile group.

They have used Naïve Bayes Multinomial and The logistic Classifier.

Observation:

The users in age group of 13-17 describe the topics related to adolescence, school activities and immature crush, the users from 23-27 age group write more about pre-marital affairs, favourite heroines/heroes and college life and the users belonging to 33-47 age group post more about post marriage life and corporate/social activities.

2.16 Identifying the Author of Email

Authors	Journal	Year	Summary
Olivier De Vel, Malcolm Corney, Alison Anderson, George Mohay	Digital Forensic Research Conference-2017	2017	**Methods:-** Support Vector Machine **Accuracy:-** Gender - 73.7%

Proposed System:

This paper identifies the Author's Gender from Email.

They have used dataset of 253 Email.

They have used Support Vector Machine Algorithm.

They have used the Stylometry approach.

Observation:

They have achieved accuracy of 73.7% for gender classification.

They have considered only English as a Language for Emails.

2.17 N-gram approach for gender prediction

Authors	Journal	Year	Summary
T Raghunadha Reddy, B Vishnu Vardhan, P Vijayapal Reddy	IEEE 7th International Advance Computing Conference-2017	2017	**Methods:-** Naive Bayes Multinomial and Logistic classifiers **Accuracy:-** Gender - 63.2%

Proposed System:

This paper proposed a model in which document weights were calculated with combination of POS N-grams and most frequent terms.

In the proposed approach three measures were used such as term frequency in individual document, term frequency in all documents of specific profile group and the TFIDF scores of term.

This experiment was carried out on the reviews domain to predict the gender of the authors.

Observation:

Naive Bayes Multinomial and Logistic classifiers achieved good results for gender prediction in reviews domain.

It is observed that the proposed approach got good results compared to BOW approach and also observed that only POS N-grams were not sufficient to improve the performance of the gender prediction.

2.18 Effects of Age and Gender on Blogging

Authors	Journal	Year	Summary
Jonathan Schler, Moshe Koppel, Shlomo Argamon, James Pennebaker	Conference of Computational Approaches to Analysing Weblogs – Stanford, California, USA - 2006	2006	**Methods:-** Support Vector Machine **Accuracy:-** Gender - 67.3%

Proposed System:

This paper shows the analysis of 300 million words to indicate differences in writing style and content between male and female blog.

This analysis is used to determine author's age and gender.

They have used 71,000 blogs.

They have used two different kinds of distinguishing features: Style-related and Content-related.

Observation:

Female bloggers use more pronouns and negation words while Male bloggers use more articles and prepositions.

Male bloggers of all ages write more about politics, technology and money.

Female bloggers discuss their personal lives – and use more personal writing style.

Chapter 3

Language Processing with Python

3. Language Processing with Python

3.1 What is NLP?

Natural Language Processing (NLP) is a branch of AI that helps computers to understand, interpret and manipulate human language.

NLP helps developers to organize and structure knowledge to perform tasks like translation, summarization, named entity recognition, relationship extraction, speech recognition, topic segmentation, etc.

NLP is a way of computers to analyse, understand and derive meaning from a human languages such as English, Spanish, Hindi, etc.

3.2 History of NLP

Here, is are important events in the history of Natural Language Processing:

1950- NLP started when Alan Turing published an article called "Machine and Intelligence."

1950- Attempts to automate translation between Russian and English

1960- The work of Chomsky and others on formal language theory and generative syntax

1990- Probabilistic and data-driven models had become quite standard

2000- A Large amount of spoken and textual data become available.

3.3 How NLP Works ?

Before we learn how NLP works, let's understand how humans use language-

Every day, we say thousand of a word that other people interpret to do countless things. We, consider it as a simple communication, but we all know that words run much deeper than that. There is always some context that we derive from what we say and how we say it., NLP never focuses on voice modulation; it does draw on contextual patterns.

Example:

Man is to woman as king is to _____?

Meaning (king) – meaning (man) + meaning (woman)=?

The answer is- queen

Here, we can easily co-relate because man is male gender and woman is female gender. In the same way, the king is masculine gender, and its female gender is queen.

Example:

Is King to kings as the queen is to_____?

The answer is--- queens

Here, we can see two words kings and kings where one is singular and other is plural. Therefore, when the world queen comes, it automatically co-relates with queens again singular plural.

Here, the biggest question is that how do we know what words mean? Let's, say who will call it queen? The answer is we learn this thinks through experience. However, here the main question is that how computer know about the same? We need to provide enough data for Machines to learn through experience. We can feed details like

Her Majesty the Queen.

The Queen's speech during the State visit

The crown of Queen Elizabeth

The Queens's Mother.

The queen is generous.

With above examples the machine understands the entity Queen. The machine creates word vectors as below. A word vector is built using surrounding words.

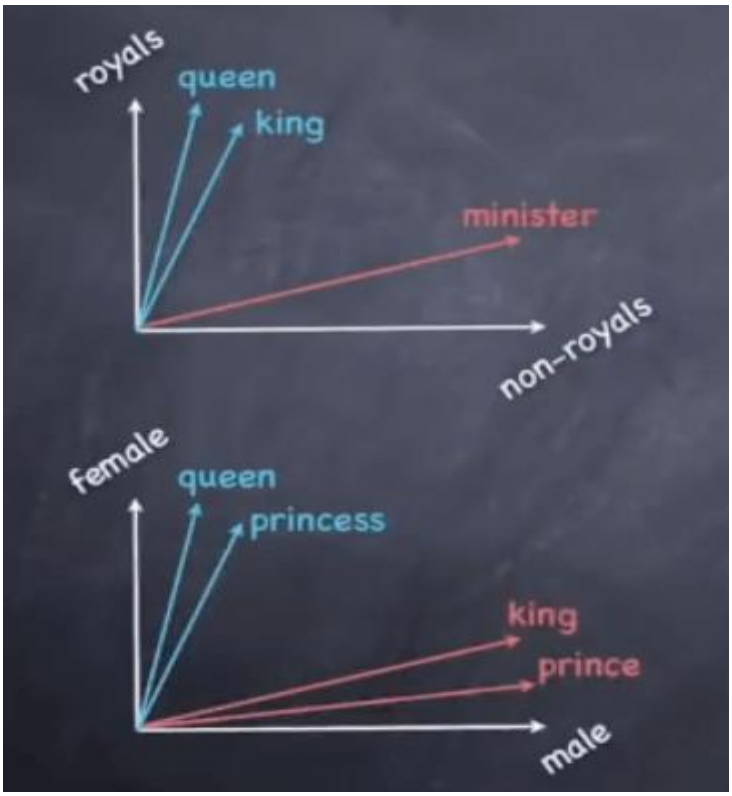


Fig 3.1 Word vector

The machine creates these vectors

As it learns from multiple datasets

Use Machine learning (e.g., Deep Learning algorithms)

A word vector is built using surrounding words.

Here is the formula:

Meaning (king) – meaning (man) + meaning (woman) =?

This amounts to performing simple algebraic operations on word vectors:

Vector (king) – vector (man) + vector (woman) = vector (?)

To which the machine answers queen.

3.4 Components of NLP

Five main Component of Natural Language processing are:

- Morphological and Lexical Analysis
- Syntactic Analysis
- Semantic Analysis
- Discourse Integration

- Pragmatic Analysis

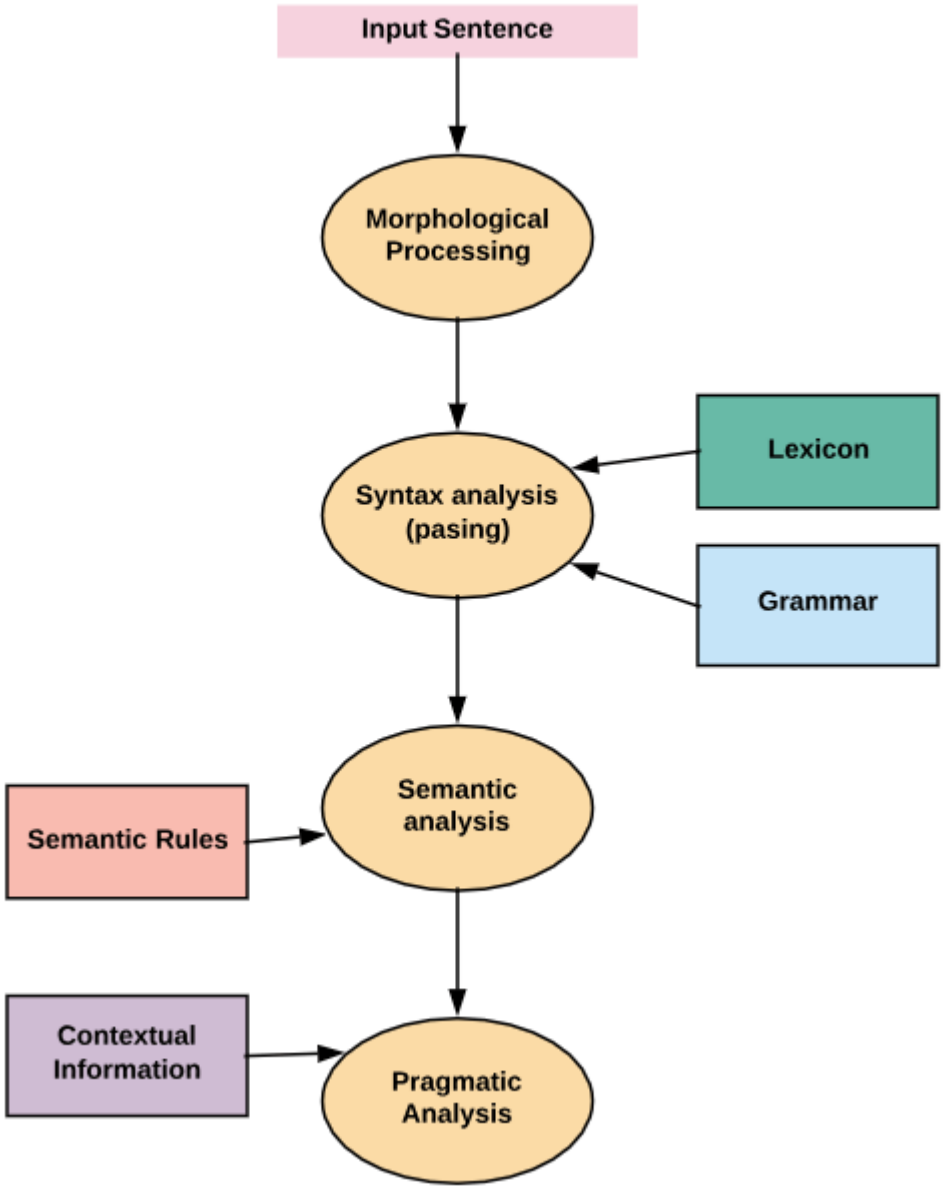


Fig 3.2 Components of NLP

Morphological and Lexical Analysis

Lexical analysis is a vocabulary that includes its words and expressions. It depicts analyzing, identifying and description of the structure of words. It includes dividing a text into paragraphs, words and the sentences. Individual words are analyzed into their components, and nonword tokens such as punctuations are separated from the words.

Semantic Analysis

Semantic Analysis is a structure created by the syntactic analyser which assigns meanings. This component transfers linear sequences of words into structures. It shows how the words are associated with each other. Semantics focuses only on the literal meaning of words, phrases, and sentences. This only abstracts the dictionary meaning or the real meaning from the given context. The structures assigned by the syntactic analyser always have assigned meaning

E.g.. "colorless green idea." This would be rejected by the Symantec analysis as colorless here; green doesn't make any sense.

Pragmatic Analysis

Pragmatic Analysis deals with the overall communicative and social content and its effect on interpretation. It means abstracting or deriving the meaningful use of language in situations. In this analysis, the main focus always on what was said in reinterpreted on what is meant. Pragmatic analysis helps users to discover this intended effect by applying a set of rules that characterize cooperative dialogues.

E.g., "close the window?" should be interpreted as a request instead of an order.

Syntax analysis

The words are commonly accepted as being the smallest units of syntax. The syntax refers to the principles and rules that govern the sentence structure of any individual languages. Syntax focus about the proper ordering of words which can affect its meaning. This involves analysis of the words in a sentence by following the grammatical structure of the sentence. The words are transformed into the structure to show hows the word are related to each other.

Discourse Integration

It means a sense of the context. The meaning of any single sentence which depends upon that sentences. It also considers the meaning of the following sentence. For example, the word "that" in the sentence "He wanted that" depends upon the prior discourse context.

3.5 NLP and Writing Systems

The kind of writing system used for a language is one of the deciding factors in determining the best approach for text pre-processing. Writing systems can be

- Logographic: a Large number of individual symbols represent words. Example Japanese, Mandarin
- Syllabic: Individual symbols represent syllables
- Alphabetic: Individual symbols represent sound

Majority of the writing systems use the Syllabic or Alphabetic system. Even English, with its relatively simple writing system based on the Roman alphabet, utilizes logographic symbols which include Arabic numerals, Currency symbols (S, £), and other special symbols.

This pose following challenges

- Extracting meaning(semantics) from a text is a challenge
- NLP is dependent on the quality of the corpus. If the domain is vast, it's difficult to understand context.
- There is a dependence on the character set and language

3.6 How to Implement NLP?

Below, given are popular methods used for Natural Learning Process:

Machine learning: The learning nlp procedures used during machine learning. It automatically focuses on the most common cases. So when we write rules by hand, it is often not correct at all concerned about human errors.

Statistical inference: NLP can make use of statistical inference algorithms. It helps you to produce models that are robust. e.g., containing words or structures which are known to everyone.

3.7 Advantages of NLP

- Users can ask questions about any subject and get a direct response within seconds.
- NLP system provides answers to the questions in natural language
- NLP system offers exact answers to the questions, no unnecessary or unwanted information
- The accuracy of the answers increases with the amount of relevant information provided in the question.
- NLP process helps computers communicate with humans in their language and scales other language-related tasks
- Allows you to perform more language-based data compares to a human being without fatigue and in an unbiased and consistent way.
- Structuring a highly unstructured data source

3.8 Disadvantages of NLP

- Complex Query Language- the system may not be able to provide the correct answer it the question that is poorly worded or ambiguous.
- The system is built for a single and specific task only; it is unable to adapt to new domains and problems because of limited functions.
- NLP system doesn't have a user interface which lacks features that allow users to further interact with the system.

3.9 What is NLTK?

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been been updated for Python 3 and NLTK 3.

3.10 Installing NLTK in Linux

Installing NLTK in Mac/Unix requires python package manager pip to install nltk. If pip is not installed, please follow the below instructions to complete the process

Step1) Update the package index by typing the below command

sudo apt update

Step2) Installing pip for Python 3:

sudo apt install python3-pip

You can also install pip using easy_install.

sudo apt-get install python-setuptools python-dev build-essential

Now easy_install is installed. Run the below command to install pip

sudo easy_install pip

Step3) Use following command to install NLTK

sudo pip install -U nltk

sudo pip3 install -U nltk

3.11 Installing NLTK through Anaconda

Step1) Please install anaconda (which can also be used to install different packages) by visiting https://www.anaconda.com/download/ and select which version of python you need to install for anaconda.

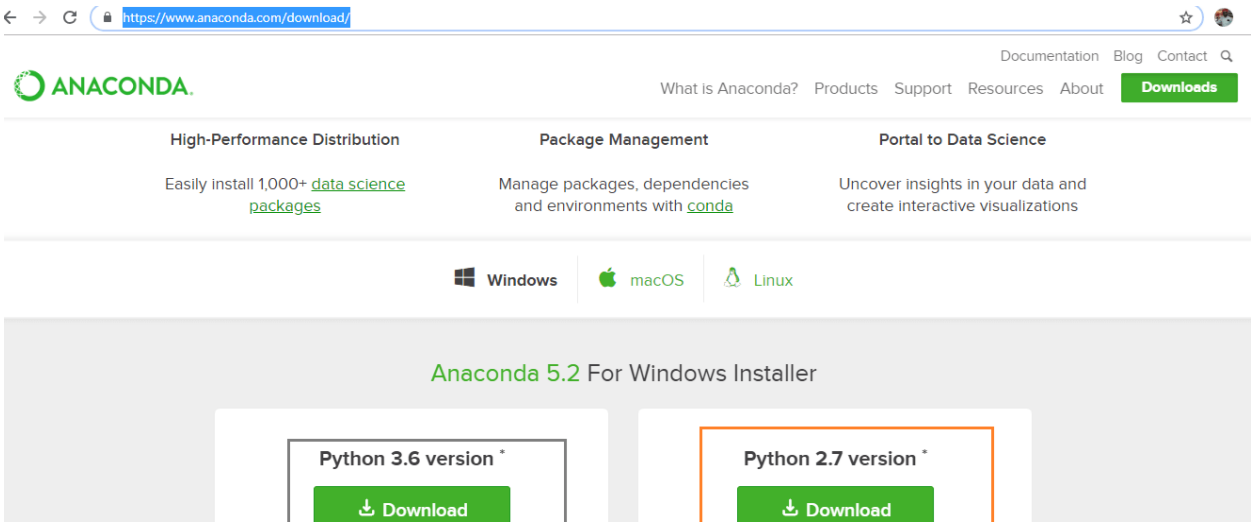


Fig 3.3 Installing Anaconda

Step 2) In the Anaconda prompt, Enter command

conda install -c anaconda nltk

Review the package upgrade, downgrade, install information and enter yes

NLTK is downloaded and installed

Fig 3.4 Installing Step

3.12 How to download all packages of NLTK?

Step 1) Run the Python interpreter in Windows or Linux

Step 2) Enter the commands

import nltk

nltk.download ()

NLTK Downloaded Window Opens. Click the Download Button to download the dataset. This process will take time, based on your internet connection.

Step 3) To test the installed data use the following code

from nltk.corpus import brown

brown.words()

['The', 'Fulton', 'County', 'Grand', 'Jury', 'said']

3.13 Tokenization

Tokenization is the process by which big quantity of text is divided into smaller parts called tokens. Natural language processing is used for building applications such as Text classification, intelligent chatbot, sentimental analysis, language translation, etc. It becomes vital to understand the pattern in the text to achieve the above-stated purpose. These tokens are very useful for finding such patterns as well as is considered as a base step for stemming and lemmatization.

For the time being, don't worry about stemming and lemmatization but treat them as steps for textual data cleaning using NLP (Natural language processing). We will discuss stemming and lemmatization later in the tutorial. Tasks such as Text classification or spam filtering makes use of NLP along with deep learning libraries such as Keras and Tensorflow. Natural Language toolkit has very important module tokenize which further compromises of sub-modules.

- Word tokenize
- Sentence tokenize

Tokenize words

A sentence or data can be split into words using the method ***word_tokenize():***

from nltk.tokenize import sent_tokenize, word_tokenize

data = "All work and no play makes jack a dull boy, all work and no play"

print(word_tokenize(data))

This will output:

['All', 'work', 'and', 'no', 'play', 'makes', 'jack', 'dull', 'boy', ',', 'all', 'work', 'and', 'no', 'play']

All of them are words except the comma. Special characters are treated as separate tokens.

Tokenizing sentences

The same principle can be applied to sentences. Simply change the to ***sent_tokenize()***

We have added two sentences to the variable data:

from nltk.tokenize import sent_tokenize, word_tokenize

data = "All work and no play makes jack dull boy. All work and no play makes jack a dull boy."

print(sent_tokenize(data))

Outputs:

['All work and no play makes jack dull boy.', 'All work and no play makes jack a dull boy.']

3.14 NLTK stopwords

Natural language processing (nlp) is a research field that presents many challenges such as natural language understanding.Text may contain stop words like ‘the’, ‘is’, ‘are’. Stop words can be filtered from the text to be processed. There is no universal list of stop words in nlp research, however the nltk module contains a list of stop words.

from nltk.tokenize import sent_tokenize, word_tokenize

data = "All work and no play makes jack dull boy. All work and no play makes jack a dull boy."

words = word_tokenize(data)

print(words)

We modify it to:

from nltk.tokenize import sent_tokenize, word_tokenize

from nltk.corpus import stopwords

data = "All work and no play makes jack dull boy. All work and no play makes jack a dull boy."

stopWords = set(stopwords.words('english'))

words = word_tokenize(data)

wordsFiltered = []

for w in words:

```
    if w not in stopWords:
        wordsFiltered.append(w)
print(wordsFiltered)
```

A module has been imported:

```
from nltk.corpus import stopwords
stopWords = set(stopwords.words('english'))
```

The returned list stopWords contains 153 stop words on my computer.

You can view the length or contents of this array with the lines:

```
print(len(stopWords))
print(stopWords)
```

We create a new list called wordsFiltered which contains all words which are not stop words.

To create it we iterate over the list of words and only add it if it's not in the stopWords list.

```
for w in words:
    if w not in stopWords:
        wordsFiltered.append(w)
```

3.15 NLTK Stemming

Stemming is a kind of normalization for words. Normalization is a technique where a set of words in a sentence are converted into a sequence to shorten its lookup. The words which have the same meaning but have some variation according to the context or sentence are normalized.

In another word, there is one root word, but there are many variations of the same words. For example, the root word is "eat" and it's variations are "eats, eating, eaten and like so". In the same way, with the help of Stemming, we can find the root word of any variations.

For example

He was riding.

He was taking the ride.

```
from nltk.stem import PorterStemmers
```

```
e_words= ["wait", "waiting", "waited", "waits"]
ps =PorterStemmer()
for w in e_words:
        rootWord=ps.stem(w)
 print(rootWord)
```

Output:

```
wait
wait
wait
wait
```

3.16 TFIDF – Term Frequency Inverse Document Frequency

Tf-idf stands for term frequency-inverse document frequency, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. One of the simplest ranking functions is computed by summing the tf-idf for each query term; many more sophisticated ranking functions are variants of this simple model.

Tf-idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

Let’s take an example to get a clearer understanding.

Sentence 1 : The car is driven on the road.

Sentence 2: The truck is driven on the highway.

In this example, each sentence is a separate document. We will now calculate the TF-IDF for the above two documents, which represent our corpus.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

Fig 3.5 TF-IDF Calculation

From the above table, we can see that TF-IDF of common words was zero, which shows they are not significant. On the other hand, the TF-IDF of “car”, “truck”, “road”, and “highway” are non-zero. These words have more significance.

3.17 What are N-Gram?

An n-gram model is a type of probabilistic language model for predicting the next item in such a sequence in the form of a (n − 1)–order Markov model. n-gram models are now widely used in probability, communication theory, computational linguistics (for instance, statistical natural language processing), computational biology (for instance, biological sequence analysis), and data compression. Two benefits of n-gram models (and algorithms that use them) are simplicity and scalability – with larger n, a model can store more context with a well-understood space–time tradeoff, enabling small experiments to scale up efficiently.

N-grams of texts are extensively used in text mining and natural language processing tasks. They are basically a set of co-occuring words within a given window and when computing the n-grams you typically move one word forward (although you can move X words forward in more advanced scenarios). For example, for the sentence "The cow jumps over the moon". If N=2 (known as **bigrams**), then the ngrams would be:

the cow

cow jumps

jumps over

over the

the moon

So you have 5 n-grams in this case. Notice that we moved from the->cow to cow->jumps to jumps->over, etc., essentially moving one word forward to generate the next bigram.

If N=3, the n-grams would be:

the cow jumps

cow jumps over

jumps over the

over the moon

So you have 4 n-grams in this case. When N=1, this is referred to as unigrams and this is essentially the individual words in a sentence. When N=2, this is called bigrams and when N=3 this is called **trigrams**. When N>3 this is usually referred to as four grams or five grams and so on.

Chapter 4

Machine Learning Approach

4. Machine Learning Approach

4.1 What is Machine Learning?

Machine Learning is a concept which allows the machine to learn from examples and experience, and that too without being explicitly programmed. So instead of you writing the code, what you do is you feed data to the generic algorithm, and the algorithm/ machine builds the logic based on the given data.

Machine Learning is a subset of artificial intelligence which focuses mainly on machine learning from their experience and making predictions based on its experience.

It enables the computers or the machines to make data-driven decisions rather than being explicitly programmed for carrying out a certain task. These programs or algorithms are designed in a way that they learn and improve over time when are exposed to new data.

4.2 How Machine Learning Works?

Machine Learning algorithm is trained using a training data set to create a model. When new input data is introduced to the ML algorithm, it makes a prediction on the basis of the model. The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set. This is just a very high-level example as there are many factors and other steps involved.

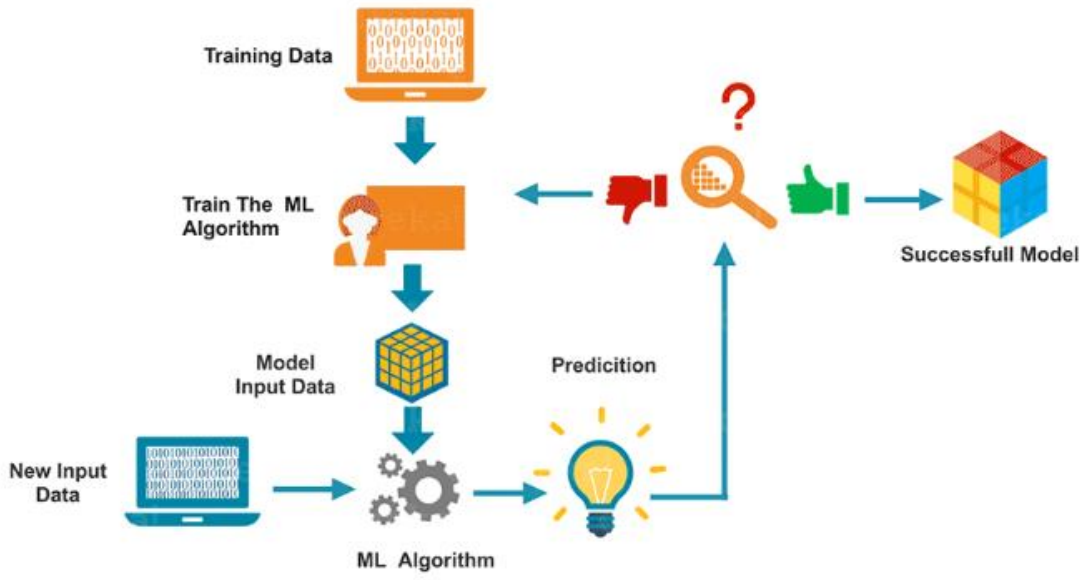


Fig 4.1 Machine Learning Working

4.3 Types of Machine Learning

- Supervised Learning – Train Me!
- Unsupervised Learning – I am self sufficient in learning
- Reinforcement Learning – My life My rules! (Hit & Trial)

4.4 What is Supervised Learning?

Supervised Learning is the one, where you can consider the learning is guided by a teacher. We have a dataset which acts as a teacher and its role is to train the model or the machine. Once the model gets trained it can start making a prediction or decision when new data is given to it.

4.5 What is Unsupervised Learning?

The model learns through observation and finds structures in the data. Once the model is given a dataset, it automatically finds patterns and relationships in the dataset by creating clusters in it. What it cannot do is add labels to the cluster, like it cannot say this a group of apples or mangoes, but it will separate all the apples from mangoes.

Suppose we presented images of apples, bananas and mangoes to the model, so what it does, based on some patterns and relationships it creates clusters and divides the dataset into those clusters. Now if a new data is fed to the model, it adds it to one of the created clusters.

4.6 What is Reinforcement Learning?

It is the ability of an agent to interact with the environment and find out what is the best outcome. It follows the concept of hit and trial method. The agent is rewarded or penalized with a point for a correct or a wrong answer, and on the basis of the positive reward points gained the model trains itself. And again once trained it gets ready to predict the new data presented to it.

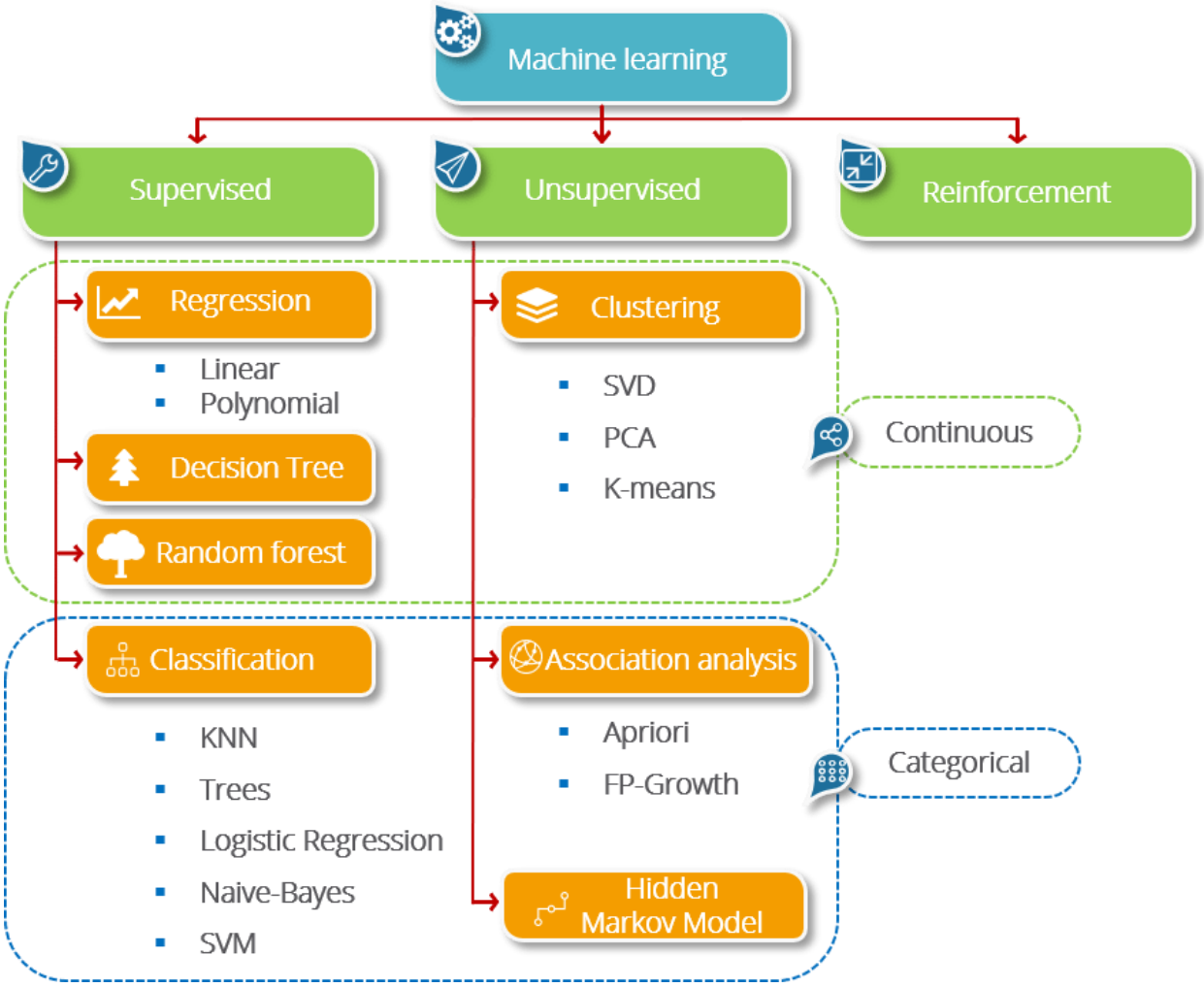


Fig 4.2 Classification Algorithm in ML

4.7 Python Libraries for Machine Learning

Machine Learning, as the name suggests, is the science of programming a computer by which they are able to learn from different kinds of data. A more general definition given by Arthur Samuel is – "Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed." They are typically used to solve various types of life problems.

In the older days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

4.7.1 Numpy

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

Example :

```
import numpy as np

# Creating two arrays of rank 2

x = np.array([[1, 2], [3, 4]])

y = np.array([[5, 6], [7, 8]])

# Creating two arrays of rank 1

v = np.array([9, 10])

w = np.array([11, 12])

# Inner product of vectors

print(np.dot(v, w), "\n")

# Matrix and Vector product

print(np.dot(x, v), "\n")

# Matrix and matrix product

print(np.dot(x, y))
```

OUTPUT :

219

[29 67]

[[19 22]

[43 50]]

4.7.2 Scikit-learn

Scikit-learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with ML.

Example :

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,

max_features=None, max_leaf_nodes=None,

min_impurity_decrease=0.0, min_impurity_split=None,

min_samples_leaf=1, min_samples_split=2,

min_weight_fraction_leaf=0.0, presort=False, random_state=None,

splitter='best')

	precision	recall	f1-score	support
0	1.00	1.00	1.00	50
1	1.00	1.00	1.00	50
2	1.00	1.00	1.00	50
micro avg	1.00	1.00	1.00	150
macro avg	1.00	1.00	1.00	150
weighted avg	1.00	1.00	1.00	150

[[50 0 0]

[0 50 0]

[0 0 50]]

4.7.3 TensorFlow

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

```
import tensorflow as tf

 # Initialize two constants

x1 = tf.constant([1, 2, 3, 4])

x2 = tf.constant([5, 6, 7, 8])

 # Multiply

result = tf.multiply(x1, x2)

 # Intialize the Session

sess = tf.Session()

print(sess.run(result))

# Close the session

sess.close()
```

OUTPUT :

```
[ 5 12 21 32]
```

4.7.4 Pandas

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

```
import pandas as pd

data = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],

 "capital": ["Brasilia", "Moscow", "New Dehli", "Beijing", "Pretoria"],

"area": [8.516, 17.10, 3.286, 9.597, 1.221],

"population": [200.4, 143.5, 1252, 1357, 52.98] }

data_table = pd.DataFrame(data)

print(data_table)
```

OUTPUT :

	country	capital	area	population
0	Brazil	Brasilia	8.516	200.40
1	Russia	Moscow	17.100	143.50
2	India	New Dehli	3.286	1252.00
3	China	Beijing	9.597	1357.00
4	South Africa	Pretoria	1.221	52.98

4.7.5 Matplotlib

Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar chats, etc,

```
import matplotlib.pyplot as plt

import numpy as np

x = np.linspace(0, 10, 100)

plt.plot(x, x, label ='linear')

plt.legend()

# Show the plot

plt.show()
```

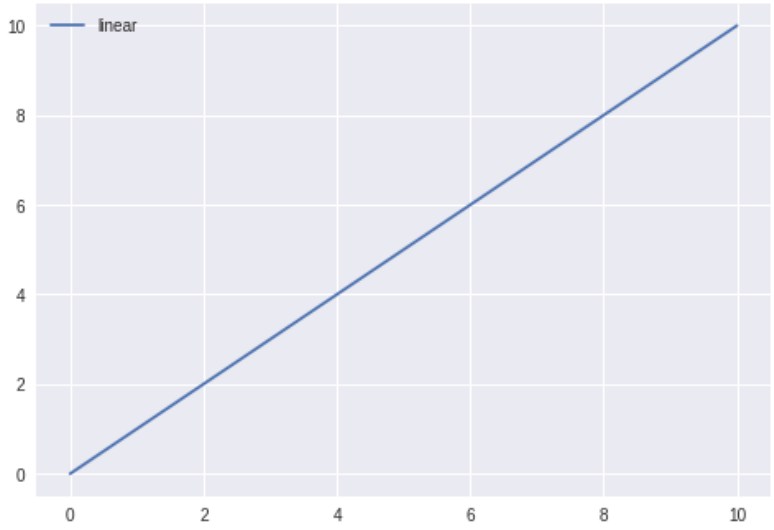


Fig 4.3 Matpoltlib vector

4.8 Machine Learning Classification

Supervised Learning is the one, where you can consider the learning is guided by a teacher. We have a dataset which acts as a teacher and its role is to train the model or the machine. Once the model gets trained it can start making a prediction or decision when new data is given to it.

Supervised Machine Learning Algorithms are

- Naïve Bayes
- Support Vector Machine
- Decision Tree
- Random Forest
- Logistic Regression etc.

4.8.1 Naïve Bayes

Naive Bayes is one of the most common ML algorithms that is often used for the purpose of text classification. If you have just stepped into ML, it is one of the easiest classification algorithms to start with. Naive Bayes is a probabilistic classification algorithm as it uses probability to make predictions for the purpose of classification.

Bayes Theorem

$$P(A|B) = \frac{P(B|A) * (P(A)}{P(B)}$$

The above equation represents Bayes Theorem in which it describes the probability of an event occurring P(A) based on our prior knowledge of events that may be related to that event P(B).

The parts of Bayes Theorem:

P(A|B) - Posterior Probability

The conditional probability that event A occurs given that event B has occurred.

P(A) - Prior Probability

The probability of event A.

P(B) - Evidence

The probability of event B.

P(B|A) - Likelihood

The conditional probability of B occurring given event A has occurred.

Type of Naive Bayes Algorithm

Python's Scikitlearn gives the user access to the following 3 Naive Bayes models.

Gaussian

The gaussian NB Alogorithm assumes all contnuous features (predictors) and all follow a Gaussian (Normal Distribution).

Multinomial

Multinomial NB is suited for discrete data that have frequencies and counts. Spam Filtering and Text/Document Classification are two very well-known use cases.

Bernoulli

Bernoulli is similar to Multinomial except it is for boolean/binary features. Like the multinomial method it can be used for spam filtering and document classification in which binary terms (i.e. word occurrence in a document represented with True or False).

4.8.2 Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes. SVMs are more commonly used in classification problems and as such, this is what we will focus on in this post. SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes, as shown in the image below.

How SVM Works ?

The basics of Support Vector Machines and how it works are best understood with a simple example. Let’s imagine we have two tags: red and blue, and our data has two features: x and y. We want a classifier that, given a pair of (x,y) coordinates, outputs if it’s either red or blue. We plot our already labeled training data on a plane:

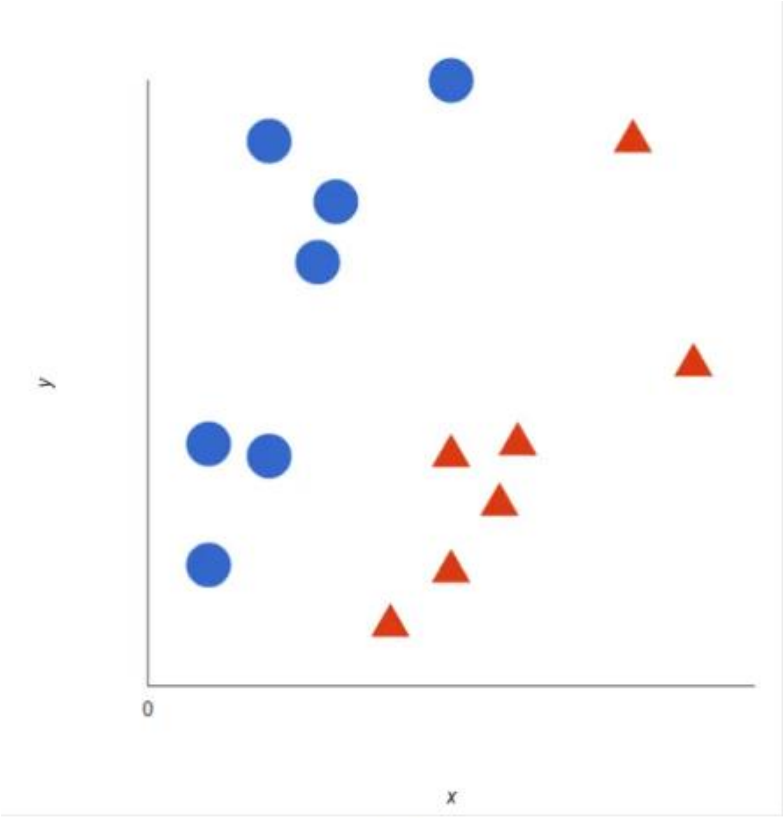


Fig 4.4 SVM - Data Label

A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it's simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red.

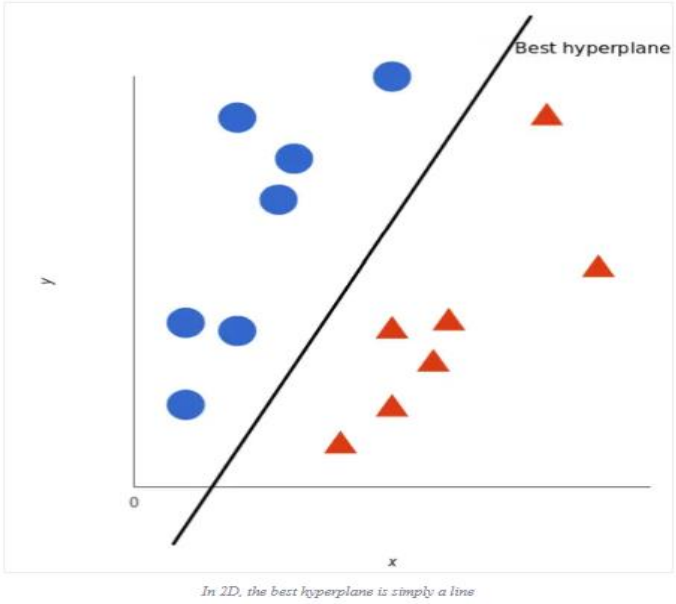


Fig 4.5 HyperPlane

But, what exactly is the best hyperplane? For SVM, it's the one that maximizes the margins from both tags. In other words: the hyperplane (remember it's a line in this case) whose distance to the nearest element of each tag is the largest.

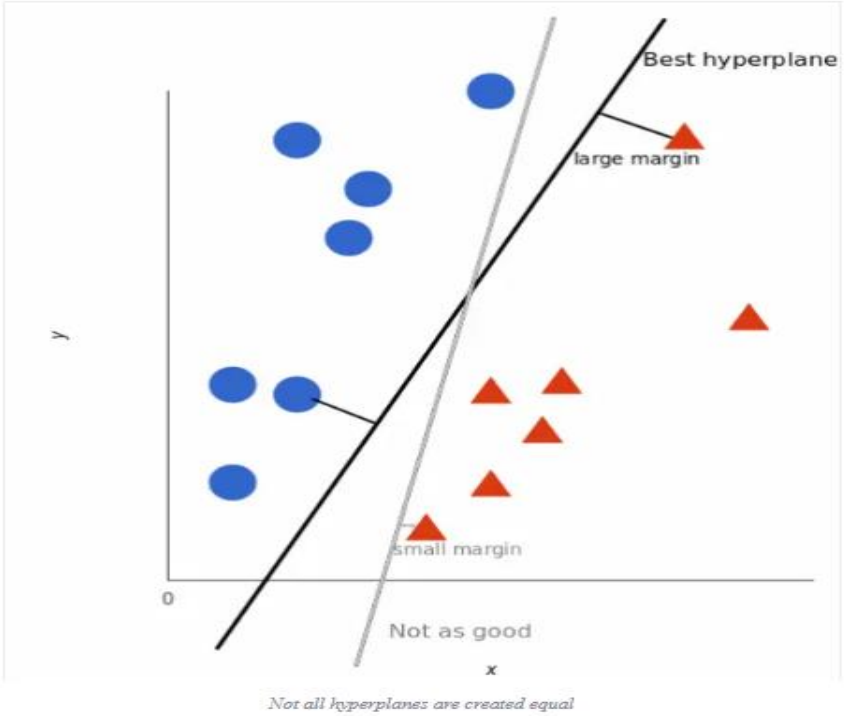


Fig 4.6 Multi-Hyperplane

Nonlinear data

Now this example was easy, since clearly the data was linearly separable — we could draw a straight line to separate red and blue. Sadly, usually things aren't that simple. Take a look at this case:

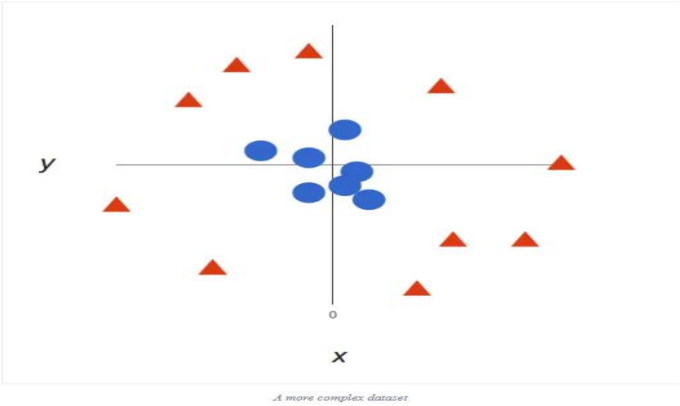


Fig 4.7 Non Linear Data

It’s pretty clear that there’s not a linear decision boundary (a single straight line that separates both tags). However, the vectors are very clearly segregated and it looks as though it should be easy to separate them. So here’s what we’ll do: we will add a third dimension. Up until now we had two dimensions: x and y. We create a new z dimension, and we rule that it be calculated a certain way that is convenient for us: $z = x^2 + y^2$ (you’ll notice that’s the equation for a circle).

This will give us a three-dimensional space. Taking a slice of that space, it looks like this:

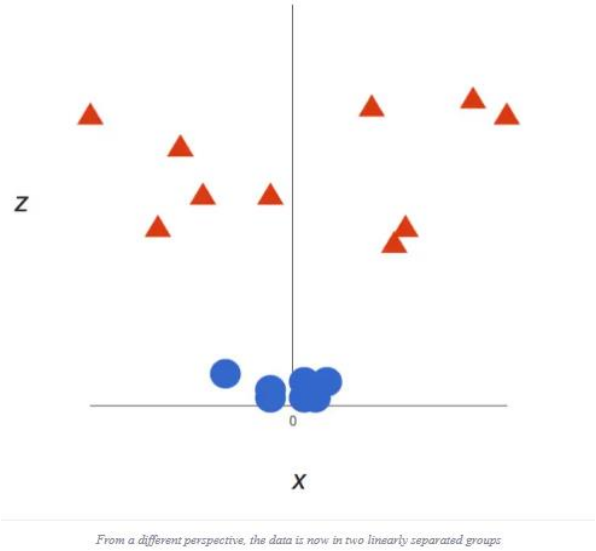


Fig 4.8 Multi Dimension in SVM

Note that since we are in three dimensions now, the hyperplane is a plane parallel to the x axis at a certain z (let’s say z = 1).

What’s left is mapping it back to two dimensions:

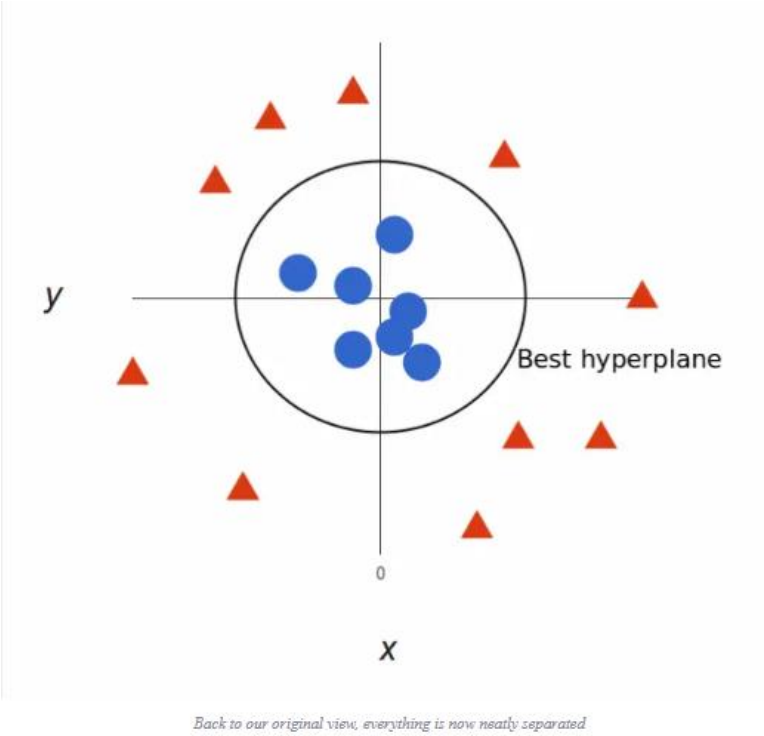


Fig 4.8 Best Hypeplane in SVM

The kernel trick in SVM

In our example we found a way to classify nonlinear data by cleverly mapping our space to a higher dimension. However, it turns out that calculating this transformation can get pretty computationally expensive: there can be a lot of new dimensions, each one of them possibly involving a complicated calculation. Doing this for every vector in the dataset can be a lot of work, so it'd be great if we could find a cheaper solution. And we're in luck! Here's a trick: SVM doesn't need the actual vectors to work its magic, it actually can get by only with the dot products between them. This means that we can sidestep the expensive calculations of the new dimensions! This is what we do instead:

Imagine the new space we want:

$z = x^2 + y^2$

Figure out what the dot product in that space looks like:

$a \cdot b = xa \cdot xb + ya \cdot yb + za \cdot zb$

$a \cdot b = xa \cdot xb + ya \cdot yb + (xa^2 + ya^2) \cdot (xb^2 + yb^2)$

Give input to SVM to do its thing, but using the new dot product — we call this a kernel function.

That's it! That's the kernel trick, which allows us to sidestep a lot of expensive calculations. Normally, the kernel is linear, and we get a linear classifier. However, by using a nonlinear kernel (like above) we can get a nonlinear classifier without transforming the data at all: we only change the dot product to that of the space that we

want and SVM will happily chug along. Note that the kernel trick isn't actually part of SVM. It can be used with other linear classifiers such as logistic regression. A support vector machine only takes care of finding the decision boundary.

How can SVM be used with natural language classification?

We can classify vectors in multidimensional space. Great! Now, we want to apply this algorithm for text classification, and the first thing we need is a way to transform a piece of text into a vector of numbers so we can run SVM with them. In other words, which features do we have to use in order to classify texts using SVM?

The most common answer is word frequencies, just like we did in Naive Bayes. This means that we treat a text as a bag of words, and for every word that appears in that bag we have a feature. The value of that feature will be how frequent that word is in the text. This method boils down to just counting how many times every word appears in a text and dividing it by the total number of words. So in the sentence "All monkeys are primates but not all primates are monkeys" the word monkeys has a frequency of 2/10 = 0.2, and the word but has a frequency of 1/10 = 0.1.

For a more advanced alternative for calculating frequencies, we can also use TF-IDF.

Now that we've done that, every text in our dataset is represented as a vector with thousands (or tens of thousands) of dimensions, every one representing the frequency one of the words of the text. Perfect! This is what we feed to SVM for training. We can improve this by using preprocessing techniques, like stemming, removing stopwords, and using n-grams.

Choosing a kernel function

Now that we have the feature vectors, the only thing left to do is choosing a kernel function for our model. Every problem is different, and the kernel function depends on what the data looks like. In our example, our data was arranged in concentric circles, so we chose a kernel that matched those data points. Taking that into account, what's best for natural language processing? Do we need a nonlinear classifier? Or is the data linearly separable? It turns out that it's best to stick to a linear kernel.

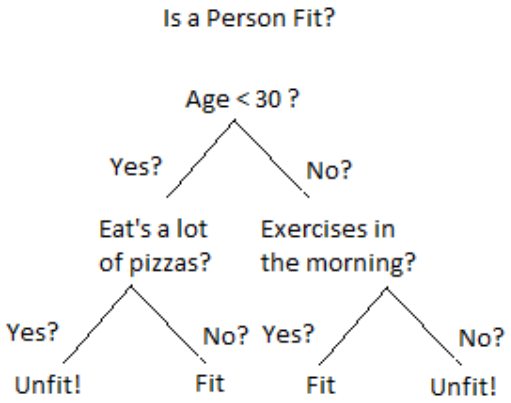
Back in our example, we had two features. Some real uses of SVM in other fields may use tens or even hundreds of features. Meanwhile, NLP classifiers use thousands of features, since they can have up to one for every word that appears in the training data. This changes the problem a little bit: while using nonlinear kernels may be a good idea in other cases, having this many features will end up making nonlinear kernels overfit the data. Therefore, it's best to just stick to a good old linear kernel, which actually results in the best performance in these cases.

A support vector machine allows you to classify data that's linearly separable. If it isn't linearly separable, you can use the kernel trick to make it work. However, for text classification it's better to just stick to a linear kernel.

4.8.3 Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.



How Do Decision Trees Work?

There are several steps involved in the building of a decision tree.

They are Splitting, Pruning and Tree Selection.

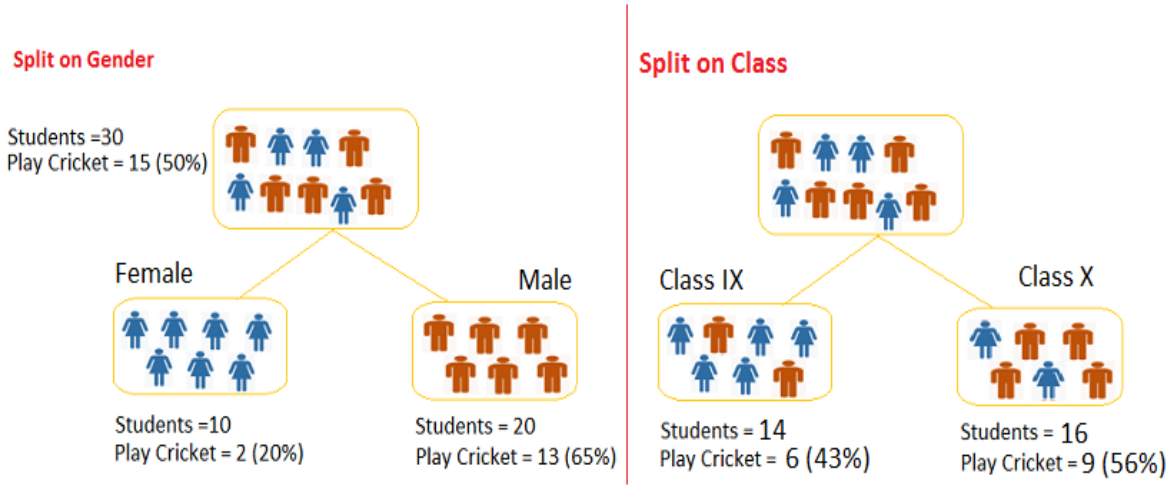


Fig 4.9 Tree Data Splitting

Splitting

The process of partitioning the data set into subsets. Splits are formed on a particular variable.

Pruning

The shortening of branches of the tree. Pruning is the process of reducing the size of the tree by turning some branch nodes into leaf nodes, and removing the leaf nodes under the original branch. Pruning is useful because classification trees may fit the training data well, but may do a poor job of classifying new values. A simpler tree often avoids over-fitting.

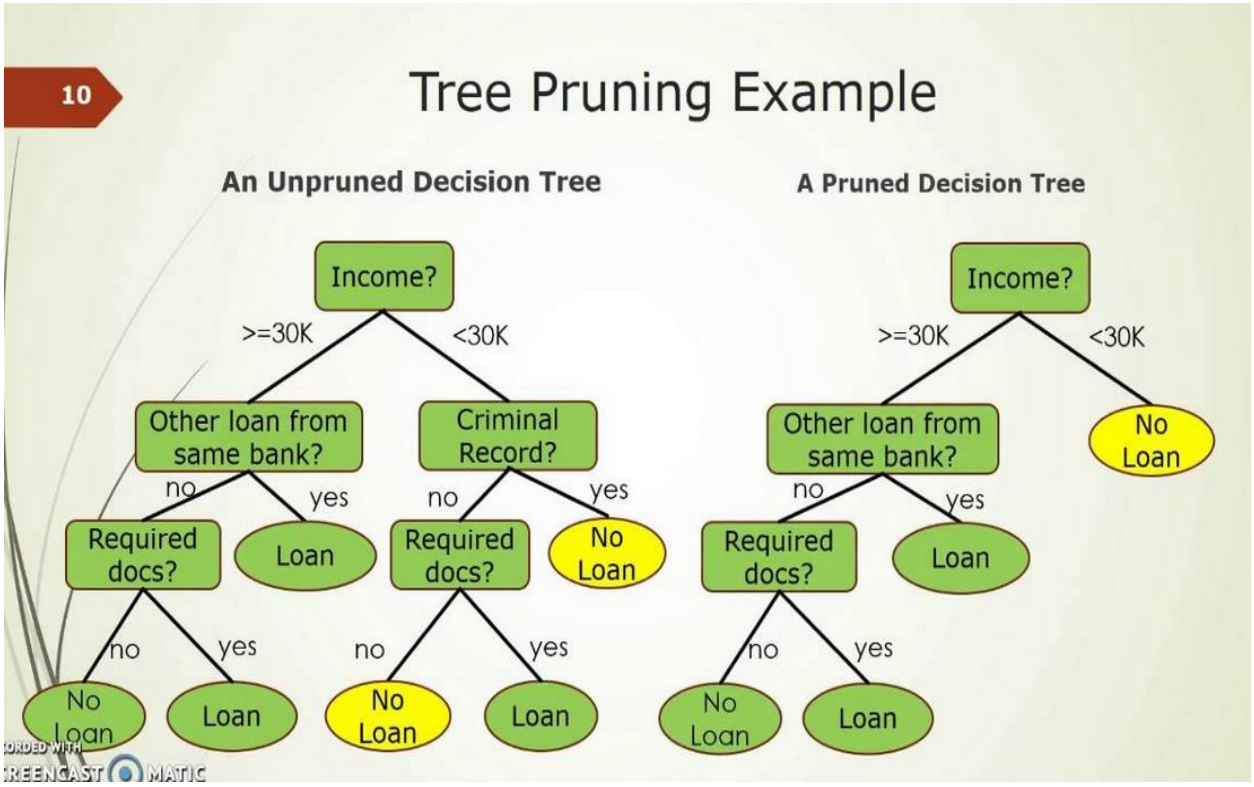


Fig 4.10 Tree Pruning

Tree Selection

The process of finding the smallest tree that fits the data. Usually this is the tree that yields the lowest cross-validated error.

Entropy

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogeneous). ID 3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.

Information Gain

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

4.9 Random Forest

The random Forest algorithm was created by Leo Brieman and Adele Cutler in 2001. Random forest is a tree-based algorithm which involves building several trees (decision trees), then combining their output to improve generalization ability of the model. The method of combining trees is known as an ensemble method. Ensembling is nothing but a combination of weak learners (individual trees) to produce a strong learner.

Say, you want to watch a movie. But you are uncertain of its reviews. You ask 10 people who have watched the movie. 8 of them said " the movie is fantastic." Since the majority is in favor, you decide to watch the movie. This is how we use ensemble techniques in our daily life too.

Random Forest can be used to solve regression and classification problems. In regression problems, the dependent variable is continuous. In classification problems, the dependent variable is categorical.

How does it works?

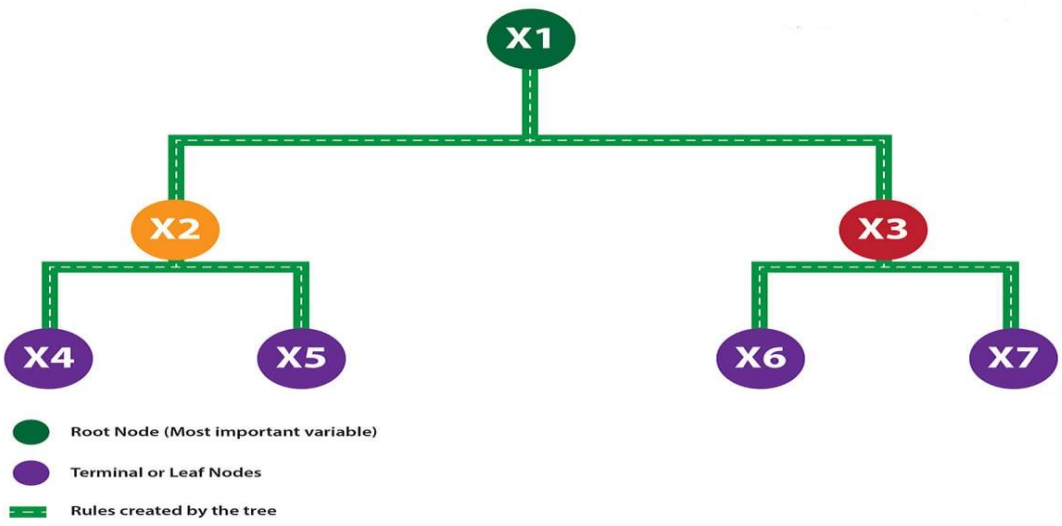


Fig 4.11 Random Forest Tree

In regression trees (where the output is predicted using the mean of observations in the terminal nodes), the splitting decision is based on minimizing RSS. The variable which leads to the greatest possible reduction in RSS is chosen as the root node. The tree splitting takes a top-down greedy approach, also known as recursive binary splitting. We call it "greedy" because the algorithm cares to make the best split at the current step rather than saving a split for better results on future nodes.

In classification trees (where the output is predicted using mode of observations in the terminal nodes), the splitting decision is based on the following methods:

Gini Index - It's a measure of node purity. If the Gini index takes on a smaller value, it suggests that the node is pure. For a split to take place, the Gini index for a child node should be less than that for the parent node.

Entropy - Entropy is a measure of node impurity. For a binary class (a,b), the formula to calculate it is shown below. Entropy is maximum at p = 0.5. For p(X=a)=0.5 or p(X=b)=0.5 means, a new observation has a 50%-50% chance of getting classified in either classes. The entropy is minimum when the probability is 0 or 1.

Entropy = - p(a)*log(p(a)) - p(b)*log(p(b))

Difference between Bagging and Random Forest?

It creates randomized samples of the data set (just like random forest) and grows trees on a different sample of the original data. The remaining 1/3 of the sample is used to estimate unbiased OOB error.

It considers all the features at a node (for splitting). Once the trees are fully grown, it uses averaging or voting to combine the resultant predictions.

Aren't you thinking, "If both the algorithms do same thing, what is the need for random forest? Couldn't we have accomplished our task with bagging?" NO! The need for random forest surfaced after discovering that the bagging algorithm results in correlated trees when faced with a data set having strong predictors. Unfortunately, averaging several highly correlated trees doesn't lead to a large reduction in variance. But how do correlated trees emerge? Good question! Let's say a data set has a very strong predictor, along with other moderately strong predictors. In bagging, a tree grown every time would consider the very strong predictor at its root node, thereby resulting in trees similar to each other. The main difference between random forest and bagging is that random forest considers only a subset of predictors at a split. This results in trees with different predictors at top split, thereby resulting in decorrelated trees and more reliable average output. That's why we say random forest is robust to correlated predictors.

Advantages are as follows:

- It is robust to correlated predictors.
- It is used to solve both regression and classification problems.
- It can be also used to solve unsupervised ML problems.
- It can handle thousands of input variables without variable selection.
- It can be used as a feature selection tool using its variable importance plot.
- It takes care of missing data internally in an effective manner.

Disadvantages are as follows:

- The Random Forest model is difficult to interpret.
- It tends to return erratic predictions for observations out of range of training data. For example, the training data contains two variable x and y. The range of x variable is 30 to 70. If the test data has x = 200, random forest would give an unreliable prediction.
- It can take longer than expected time to computer a large number of trees.

4.9.1 Logistic Regression

It's a classification algorithm, that is used where the response variable is categorical. The idea of Logistic Regression is to find a relationship between features and probability of particular outcome. When we have to predict if a student passes or fails in an exam when the number of hours spent studying is given as a feature, the response variable has two values, pass and fail.

This type of a problem is referred to as Binomial Logistic Regression, where the response variable has two values 0 and 1 or pass and fail or true and false. Multinomial Logistic Regression deals with situations where the response variable can have three or more possible values.

Why Logistic, not linear?

With binary classification, let 'x' be some feature and 'y' be the output which can be either 0 or 1.

The probability that the output is 1 given its input can be represented as:

If we predict the probability via linear regression, we can state it as:

where, $p(x) = p(y=1|x)$

Linear regression model can generate the predicted probability as any number ranging from negative to positive infinity, whereas probability of an outcome can only lie between $0< P(x)<1$.

Performance of Logistic Regression model:

To evaluate the performance of a logistic regression model, Deviance is used in lieu of sum of squares calculations. Null Deviance indicates the response predicted by a model with nothing but an intercept.

Model deviance indicates the response predicted by a model on adding independent variables. If the model deviance is significantly smaller than the null deviance, one can conclude that the parameter or set of parameters significantly improved model fit.

Another way to find the accuracy of model is by using Confusion Matrix.

The accuracy of the model is given by:

$$\frac{\text{True Positive + True Negatives}}{\text{True Positive + True Negatives + False Positives + False Negatives}}$$

Fig 4.12Linear Regression Accuracy

Chapter 5

Proposed Model

5. Proposed Model

Multilingual Author Profiling on SMS aims to predict the author's age and gender from the messages. Here the messages are in Roman Hindi and English. Messages has become a popular medium of communication which is very useful for author profiling. Most research has so far concentrated on English texts; however, more than half of the users are writing in other languages. Many users often change the language while posting on social media which is called code-mixing, and it develops some challenges in the field of text classification and author profiling. Here we analyse the task of author's gender and age prediction in code-mixed content and present a corpus of English-Hindi texts collected from FIRE and by us which is annotated with author's gender and age. We present a supervised classification system which uses various machine learning algorithms to identify the gender and age of an author.

5.1 Research Problem

We aim to find the Author's age and Gender from the messages which are written in Roman Hindi and English.

Gender Identification:

To classify the multilingual author to Male or Female.

Age Identification:

To classify the multilingual author profile into one of the three categories. 15-19, 20-24, 25-XX.

Multilingualism is the use of more than one language, either by an individual speaker or by a community of speakers. It is believed that multilingual speakers outnumber monolingual speakers in the world's population. More than half of all Europeans claim to speak at least one language other than their mother tongue. Using more than one language in communication.

A text Classification technique that is used to predict the profiling characteristics of the authors like gender, age, native language and educational background by analyzing their text."

Finding demographic features like age, gender, native language of an author from the written text. We will be using Roman Hindi and English as our Multilingual Data. Roman Hindi (written using English Alphabets) is used in daily communication by a large number of people. It is used in comments, Tweets, Blogs, SMS -messages etc.

Majority of the research on Author Profiling is done for English, Spanish, Italian and Dutch Language.

5.2 Applications

- It can be used in forensics to find the suspect.
- It can be used to find out the Fake Profile Identification.
- It can also be used in security and marketing.

5.3 Dataset Classification

We will be using the Dataset provided by the FIRE2018.

Age Group	Gender		TOTAL
	Male	Female	
15-19	70	38	108
20-24	112	64	176
25- XX	28	38	66
TOTAL	210	140	350

Fig 5.1 Dataset Classification

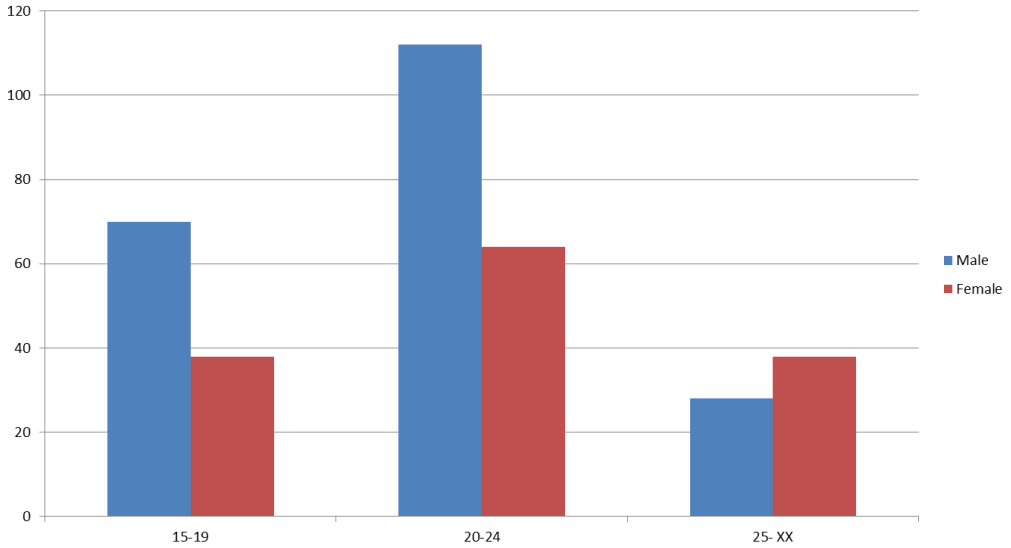


Fig 5.2 Dataset Graph

5.4 Sample Data : Male

```
Abhi tak to ni hai
Check kr rahe hain ameer sahab
Aa ni rahe?
Ma'am I'll ask him if he can come since he is doing his internahip
Ma'am what type of data do u need cleaning?
OK maam I don't know much about these things but I'll come to your office tomorrow
Ok ma'am
ROK jaa poch ke batata hn
Mein free hon aja
Uni jana hai?
Raat tak batata hn abhi office mein hn
Ajaounga kal mein. Baat krli hai
Fb bhi check krli
Aoa. Usama kya haal chal? Phr kaisa raha result? Hafiz sahab ne kya grade diya??
```

Fig 5.3 Sample Data For Male

5.5 Sample Data : Female

```
Api hamary next week ka b yehi timetable hoga????????
Sprite ly kr lab mai a jao
Laptop ka charger he nae hai :(
Tu TV deakh kiya kro movies deakh liya kro
Main nashta kr k a jao phir krna
Kaha gaeb ho aaj kal baat he nae hue kafi time ho gya
Tumary pass kon sa num hota hai???
Mujhy b bht pari thi dant :p
Dant tu nae pari ghar walo sy???
Han g kro kro mazy :)
Ghar mai sab kaisy hai???
Theak ja rahe hai. Abi tu start hai easy lag raha hai bs math Thori c Mushkil lag rahe hai
Phir b batao. Mery b bury thy
Han main soch rahe thi k Taira or fiza log chaly gye hai Aab enjoy kr rahy ho gye
Main samji Shaiyad pata ho frnds sy pata laga ho
```

Fig 5.4 Sample Data For Female

5.6 Tags Identification

We have identified following tags from our corpus.

i. Roman Hindi

The example of Hindi based words are “kya”, “raat”, “din”, “subha” and so on. English words are part of the Hindi vocabulary as well, sometimes in same meanings and sometimes in different meanings.

For Example,

‘main jaon?’ where ‘main’ is an English word, however, it is representing a Roman Hindi word in this sentence.

ii. English

Example: ‘Reached’, ‘Home’, ‘While’ and so on. Abbreviation refers to a short form of a word or phrase.

Example: ‘jk ’ can be used as descriptor in place of ‘just kidding’, ‘btw’ can be used as abbreviation in place of ‘by the way’, ‘DIY ’ for ‘do it yourself ’ etc.

iii. Named Entity

The names of people and places belong to this category.

Example: ‘Karan ’ is a masculine name, ‘Kareena’ is a feminine, name, ‘Jawahar Chowk’ is a place name, ‘Surat’ is a city name, ‘C6’ is the room number in block C, ‘GU’ is abbreviated form of Gujarat University name etc.

iv. Numeric

Date, time and number strings are included in this category.

Example:

- ‘5pm’
- ‘10-May’
- ‘6/10 ’
- ‘3214504500 ’ etc.

v. Punctuation

This denotes the punctuation symbols such as ‘?’, ‘.’, ‘,’,‘! ’ and so on. The repetition of punctuation marks is also included in this category.

Example:

- ‘enjoying?????’
- here ‘?????’ etc.

vi. Symbol

In this, special characters and their repetitions are included.

Example :

- ‘$$$$’
- ‘#’
- ‘@’ and so on.

vii. Expression

Sound expressions are distinguishing features of SMS language. Which refer to natural or sometimes non-human sounds representing expressions.

Example :

- ‘hahaha’ and ‘hehehe’ are used to represent laughter and joy.

viii. Emoticons

It refers to the facial expressions of human beings

Example :

- ‘ : (’ For sadness
- ‘ :) ’ For smile
- ‘ o.O ’ or ‘ :O ’ For wonder and so on

5.7 Different approaches

Roman Hindi -> Hindi

API and Libraries

Step 1

Transliterate API for Python

Step 2

Natural Language Tool Kit(NLTK)

Shallow Parser - Tagger

CRF-Suite

Roman Hindi -> English

Roman Hindi API and Libraries

Step 1

Google-translate API

Step 2

Natural Language tool-kit

Stanford’s Core NLP Suite

Apache Luecene

Apache OpenNLP

Roman Hindi

API and Libraries

Step 1

We don’t need to translate our data in any language.

Step 2

Natural Language tool-kit

Stanford’s Core NLP Suite

5.8 Work Flow

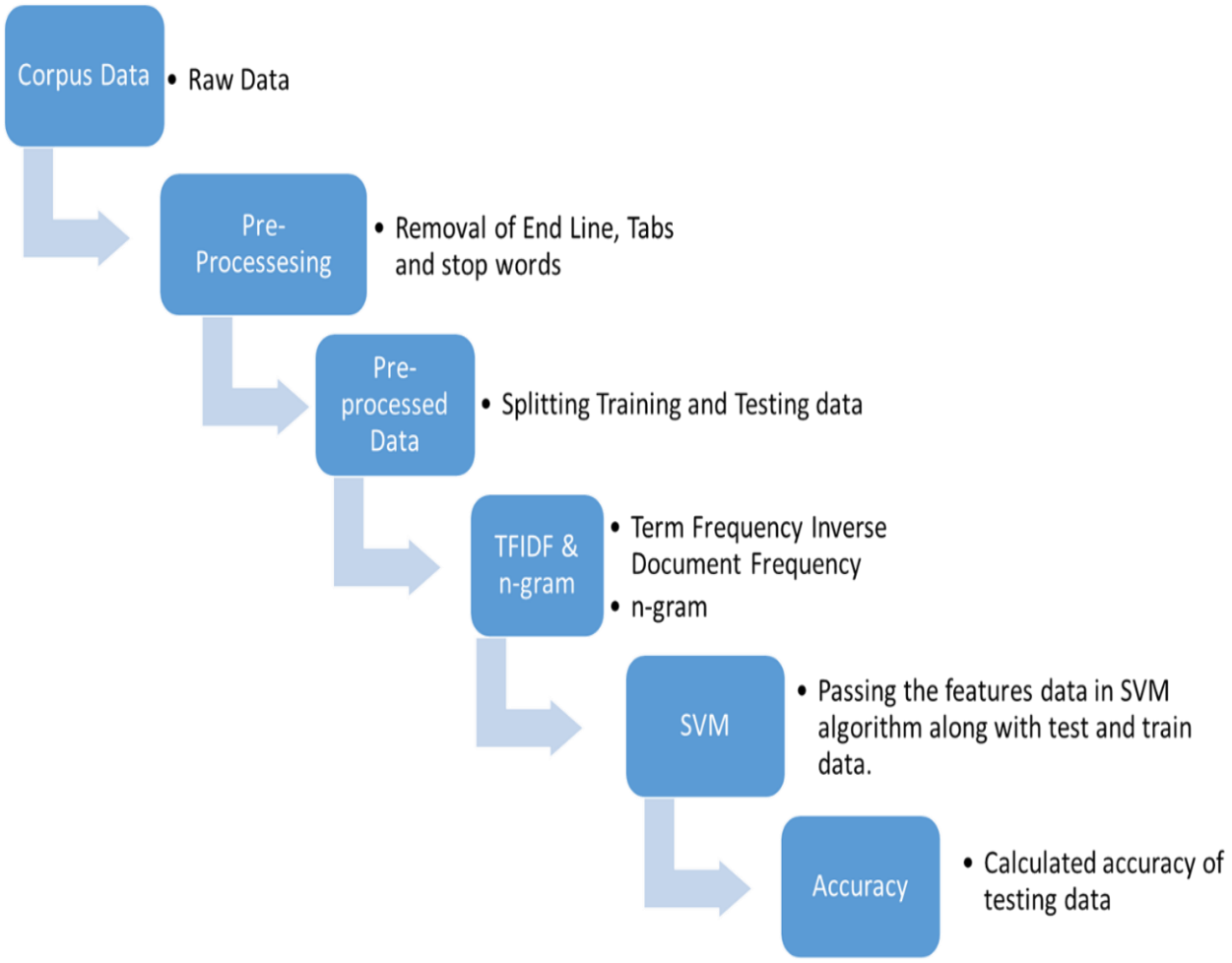


Fig 5.5 Workflow

Raw Data

parh rai thi abi khana kha k hti uphone kiii sim ko cut kr k dalnaaa paryy ga mobile may kl card lia tha or ag khtm b ho giyaa
u ny kiya socha ??? kry ga use k nae np karain bt then tata kr dain gy or pori class ko usko aik word b nae ata :-p bs aik sub
ject hai siftware engineering bs dua kr concept paky rahain and logic chl jay... mama ko bta dia hai k samosy khay hain bhook
nae hai or tu or u sy dant bhi khaiyaaa karni hai... :-p wo nano k hain u knw kisi ny darwaza nae khola mai neighbour may ai
hon :-(acha chorain sb sochna... kuch nae hota hm 3 lrkion ny old campus a k barhy waly gram garam 2 samosy liy hain now khai
ng ;-) beth gai hon last seat lmbiii waliii full akailiii bethii ;-) mai busy hon :-(ap idhr a jao photocopy shop k samny benc
h py ab mai bus may bhi 1.5 hour parhon gee or uni ja k b 10 tk free hon tb b parhon gee yani total 3.5 hour so cover ho jay ga
rat ka kam :-p :-p oke oke kl rat 4 bjy uth k parha thaaaa tuuu bhool giya kiyaaaa dil bht dukhiiii howa hai :'(:-(rona a ra
h esley jana okee i am okay. hosh may hon..or mai phr bt krtiii or sary reply daitii ..... aw aw pouch ly kisi sy download k
rna u knw mai jahan bethi hon usky pechy aik lmbiiii si seat hai last wali wahan 5 lrky hain unho ny kawaliaan lgai hoi hain or
mai sun rai bht he mazay kii hain yr achaa lg ra sun na..... done kr and i love this name ;-) nae :-p sun kon hai u ka or b
aji ka mutual frnd hai fb py back sy idea nae hotaa na ky mai aaaa raha hon ky nae :-p okay yar mera front cam mazay ke pict
ure nae laitaaa :-(:-(:-(:-(department , staff and institute ko bhii as a entity lainaa hai ..... case study b sen
d krtii hon and erd bhiii ab ankhain hain nazar pary gee to daikhon geee naaaaaaaa moving from old campus.... :-p or sardian
aa gai hain andhaira ho bhi giya bs aik do points clear nae thy wo mainy class fellow sy pouch liy thy moja kiya yr mai to soch
rai thi k ag ho jata quiz ... tiyarii bhi theek thii or tension utar jatiii .... ab tuesday tk tension rahy ge :-p yan aisa ka
ro saturday ko kr lo 8 sy 10 free hon gee mai saturady ko srs document bna rai hon software engineering kiii bnaa k u ko send k
aron geeeee :-p 100 rupees k thy jo tu ny zaya krwaaa diyyyy hain kun pehly btana tha k photocopy na krwao ainda nae kah
on ge balanxe nae hai warna mai call kr k tujhy dhoondhtii ye kia herkat thi :-[mainy tm sy poucha pechy kun bethi tm ny sb k
samny bol dia tm agy beth jao ohooo mainy department k area may a k msg parhaa hai tension ki waja sy mainy qui he nae yad kiaa
ho he nae raha tha itni tension thi but thanks god mil gay yr ab sara rm he sy note krna pary ga mujhy kun k meraa to sara g
um giya hai or tmhy achii trhaa pata hai k mai hr point khud note krtiii hon.... mujhy tasaliii nae hotiii kisi sy note krny m
ay :-[mainy tmhy kitna kahaa tha na k binii dihan sy mery pages bouht kharab howy howy hain registr k :-(or agr nae rahy to i
ska mtlb hai unkii pin utar gai thi or photocooier shop py he gir gay hain wo mairy register py kuch pages steppler howy howy t
hy....ab wo us py nae hain kahen tmhary pas to nae reh gay ??? tm ny bna li thi diagram sir sy data base ki assignment wali ??
maira registere ly aoo agr nae photocopy howa ho ga tb bhi ly ana.....bakii kl krwa laina... or aa k dy do english ki assignm
ent sy file cover utaar lana tm apna.... because mairy ko to slide nazae hee nae a rai website py :-p hiriii sun :-
p so subha py kaam na chorii chl oky jb bnay gee to mjhy ye clear kr diii k department , institute , and staff ko as a enti
ty use krnaa k nae agr rukti to phr mai 7 bjy ghr pounchtiii or itna late aa ky parhny kii hemat nae rehtii yaraaaa :-(means l
ab ka class work hee prepare karon naa mai ?? bat sun.... data base k quiz may lab ka kaam yaad kr k aana hai yan class k lect
ure ka... i mean kiya tiyarii krnii hai yr word may extra pages ko delete kaisy krty yr mujhy achaaa nae lga yr jo aj how
a..... us wakat sb thy to mainy baat khatm kr diiii.... par chalo yr agr tujhy buraaa lga tha to i appologize ... trust me mai
ra intention wo nae tha jo tu samjhiii hai....or yr ab kafii time ho giya hai hamy so yr nature samjh ja k kon kaisaa hai yr u
took me wrong....chl ainda dihan rakhon geee... now mood set kr lii...acha nae lgta friends may aisy issues hon agr.. :-) oye k

Fig 5.6 Raw Data

5.9 Tokenization

Tokenization is a step which splits longer strings of text into smaller pieces, or tokens. Larger chunks of text can be tokenized into sentences, sentences can be tokenized into words, etc. Tokenization is also referred to as text segmentation or lexical analysis. For our task, we will tokenize our sample text into a list of words. This is done using NTLK's word_tokenize () function.

Fig 5.7 Toenization

5.10 Pre-processing of data

Pre-Processing of Data (stop words removal)

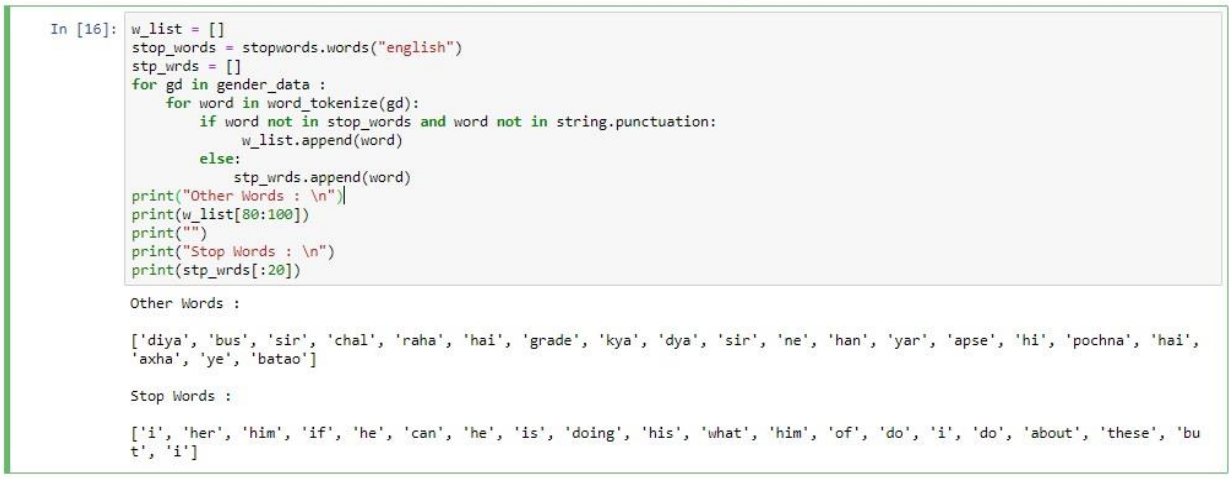


Fig 5.7 Pre-processing data-1

```
#Remove non-ASCII characters
new_words = []
for word in gender_data:
    new_word = unicodedata.normalize('NFKD', word).encode('ascii', 'ignore').decode('utf-8', 'ignore')
    new_words.append(new_word)

gender_data = new_words
new_words = []

#Remove punctuation
for word in gender_data:
    new_word = re.sub(r'[^\w\s]', '', word)
    if new_word != '':
        new_words.append(new_word)

gender_data = new_words
new_words = []
#for stopwords
for word in gender_data:
    if word not in stopwords.words('english'):
        new_words.append(word)

gender_data = new_words
new_words = []
#Lowercase
for word in gender_data:
    new_word = word.lower()
    new_words.append(new_word)
gender_data = new_words
```

Fig 5.9 pre-processing of data-2

5.11 Pre-Processed data

```
'iabhi tak to ni hai check kr rahe hain ameer sahab aa ni rahe maam ill ask him if he can come since he is doing his internahip
maam what type of data do u need cleaning ok maam i dont know much about these things but ill come to your office tomorrow ok m
aam rok jaa poch ke batata hn mein free hon aja uni jana hai raat tak batata hn abhi office mein hn ajaounga kal mein baat krli
hai fb bhi check krli aoa usama kya haal chal phr kaisa raha result hafiz sahab ne kya grade diya bus sir chal raha hai grade k
ya dya sir ne han yar apse hi pochna hai axha ye batao ke fyp 1 ka reslt 2 ke sath hi ata hai kya saleem yar mein aj ni askta e
mergency hogai hai qamar sahab ko bata di bhai ka accident hogya hai behtr hain ghr agye hain han aounga aaj kal site pe gai th
e sir qamar ko bta dya tha kaha gai the sahi sir kuch keh to ni rahe the kaha hai gya wa hn mein abhi site pe hn agr free hogya
to bata dounga warna mjhe muslim town se pic krli jub bulaoga niklne se phele mjhe bata di mein free hoga hn yaha site se bus a
rha hn office kaha hai hogya free kaha hai wapis ni ana kya woh to ni arha soya wa hai woh ni hil raha to kya krn abu tyar ni h
ain abi ok aja dehan se sir se baat hoi lame chuss na maar y  axha rokja krwa deta hn oye load mila kab tak arhi hai oye ghalat
nmbr pe hoga ab ghr aja ghr ake krwa dounga load ghr pe hi hn meine ghalat nmbr de dya tha phr tere purse se 50 rupee lke dobar
a krwa deta hn load abu ke pass ab paise ni hain haha dramay na kr ab ammi se paise leli aur bata load chahye ke ni ahsannn sir
se baat hoi mere se ok hain 50 agya tere paise bach gai phela wala ni hua tha y  axha batata hn rok ja zara arha hn mein paise
de mar mat remote khrb hai mein ab jb msg krnga to mjhe muslim town se lne ai dramay na kari magrib parh kar mein niklnga tab t
o uthe ga na lene ajai aram se to bhola ku  hmm ajao ga hn kya hai bhol na jai mjhe aaj theek hai oye tu office se color print
nikal sakta hai dafaa hi hoja tu oye ni yar apnaa email bata ek file bhejo ga uska print nikal di mail dekh mil gai print nikal
ke rakhle mein aaj office jaoga mjhe bhool na jai hn cantbzy wats matter kaise paise sorrynot possible wapsi le jawad ke sath a
ounga to leta aounga uske pas paise hongain kis tarha ka chahye bholi na move mjhe bata mummy se pochle kuch aur to ni chahye o
k axha black kaisa hona chahye ni iqbal book center pe hi theek hai smjh agai pagal horhi hai na ab jawad ke pass itne paise ho
n bhi ya ni ye pata hai dekhlounga mein hoja shokhi hoja aur gaaliyan sun ni use mein to ni ja raha aur tjhe chahye bhi sub abi
hoga na y  kehdounga usko mere masla ni hai paise uske lagne hain  kehdounga mein usko axha kehdounga kub tak hoga free ok aur
kitni dair mere pass paise ni hain aaj aur ami ne rastay se cheezen lanay ka kaha bhi hai shukr agya mein bhi kdr hai'
```

Fig 5.10 pre-processed data

5.12 Tokenization after pre-processing

Tokenization

In [5]:
```
txt_files = glob.glob(pattern)
save =[]
for txt in txt_files :
    f=open(txt)
    file_read=  f.read()
    tokens = [w for w in word_tokenize(file_read.lower())
                  if w.isalpha()]
    no_stops = [t for t in tokens if t not in stopwords.words('english')]
    save.append(no_stops)
se = pd.Series(save)
print(se)
```

```
0      [tak, ni, hai, check, kr, rahe, hain, ameer...
1      [api, hamary, next, week, ka, b, yehi, timetab...
2      [han, raha, bhai, kiyo, raha, hai, kiya, kam, ...
3      [dear, r, u, home, going, days, progress, kiun...
4      [ok, tak, aoun, kitna, paisa, dana, han, yar, ...
5      [oper, hi, houn, tokhar, pe, houn, bs, mint, a...
6      [pta, nae, msg, aya, uni, sa, means, kal, jana...
7      [han, aik, week, ki, bat, dnt, wrry, upset, ni...
8      [zarf, b, bra, rakhna, hai, ap, ne, koi, baat,...
9      [checked, mail, plz, go, final, comments, aaj,...
10     [parh, rai, thi, abi, khana, kha, k, hti, upho...
11     [yar, pata, nhi, oka, yar, shukar, hay, allah,...
12     [hmmmm, abhi, stairs, se, itni, zor, ka, gira,...
13     [pouchna, date, change, ni, hoi, form, fil, kr...
14     [friday, ko, bjy, pak, thé, match, yahooooooo, ...
15     [aoa, kia, hal, ha, jnb, ka, kidr, cl, pkg, ha...
16     [haloooo, u, ap, na, nashta, kar, liya, h, kya,...
17     [kaisa, hai, bhai, kya, kr, rha, hai, kal, ka,...
18     [jese, e, phnchu, ge, tmhe, mssge, kr, du, ge,...
19     [sunao, kia, baat, hay, p, may, abi, ghar, may...
20     [abhi, classes, le, k, hon, sath, thi, huda, s...
```


Fig 5.11 Tokenization after pre-processing

5.13 Tokenization after pre-processing

Term frequency counts the number of occurrences of term in a text document.

Mathematically it can be represented as:

Term _ Frequency _ Wij = tfij

where, tfij as the frequency of term i in document j

Term Frequency(TF) measures how frequently a term occurs in a document.

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

Total Words in our Corpus –> 2,41,489

5.14 Term Frequency

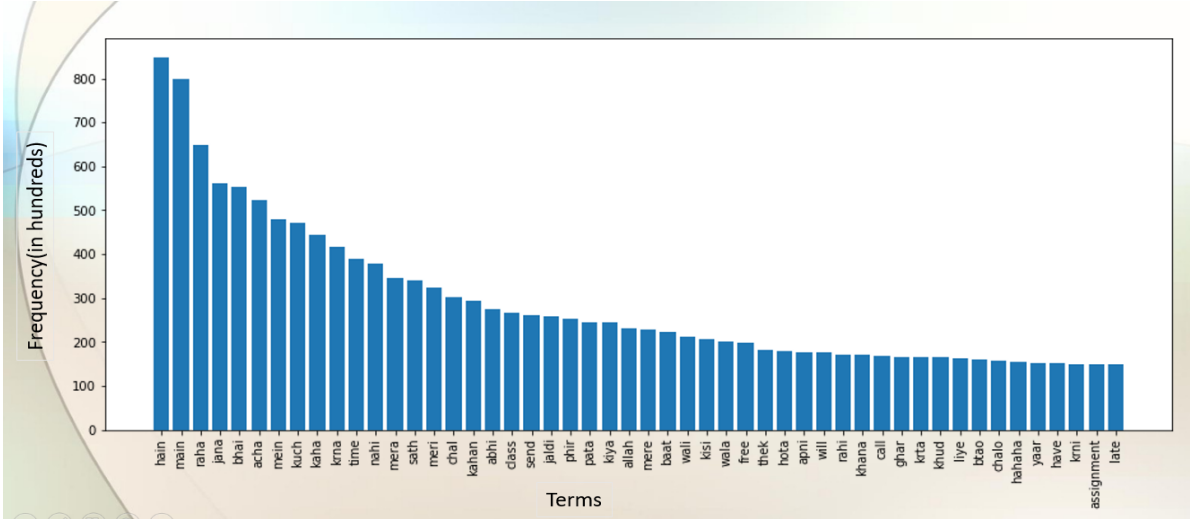


Fig 5.12 Term Frequency

5.15 TFIDF – Dataset

```
('sceen', 0.020176560431284456),
('ty', 0.026961909199138583),
('l0ck', 0.0999473294079698),
('s0ch', 0.028438750183529864),
('pheir', 0.023038568937651 67),
('singh', 0.020180396255349987),
('daily', 0.030520240643784897),
('scen', 0.0258163823361858),
('kpr', 0.030520240643784897),
('bf', 0.01527135917545028),
('shukkr', 0.030520240643784897),
('saasu', 0.014099573497550 87),
('theik', 0.03689850177617939),
('grip', 0.0258163823361858),
('background', 0.024089072792991034),
('isa', 0.028438750183529864),
('10mint', 0.0516327646723716),
('lny', 0.028438750183529864),
('ix', 0.030520240643784897),
('achaaa', 0.030520240643784897),
('shahbg', 0.07105502339836371),
('akla', 0.026961909199138583),
('tumnay', 0.061040481287569795),
('enquequ', 0.03739943889378635),
('original', 0.028438750183529864),
('sawal', 0.030520240643784897),
('dokoi', 0.030520240643784897),
('imaa', 0.024089072792991034),
('told', 0.0258163823361858),
('token', 0.026961909199138583),
('rukoon', 0.028438750183529864),
('tuje', 0.02176876777923042),
('tny', 0.024089072792991034),
('stopprtndng', 0.02279892827862852),
('arshu', 0.026961909199138583),
('ubcle', 0.16382504428144587),
('soo', 0.016286914865199648),
('musibat', 0.024880418738883553),
('lin', 0.030520240643784897),
('awi', 0.034318212747454356),
```

Chapter 6

Experiments and Results

6. Experiments and results

6.1 Naïve Bayes

- A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task.
- A naive Bayes classifier is an algorithm that uses Bayes' theorem to classify data.
- Popular uses of naive Bayes classifiers include spam filters, text analysis and medical diagnosis.

$$P(A|B) = \frac{P(B|A) * (P(A)}{P(B)}$$

The parts of Bayes Theorem:

P(A|B) - Posterior Probability

The conditional probability that event A occurs given that event B has occurred.

P(A) - Prior Probability

The probability of event A.

P(B) - Evidence

The probability of event B.

P(B|A) - Likelihood

The conditional probability of B occurring given event A has occurred.

Naïve Bayes Results			
	Gender	Age	Combine
Unigram	80.00	57.14	38.28
Bigram	65.71	50.00	27.14
Trigram	60.00	45.71	27.14

Table 6.1 Naïve bayes results

6.2 Support Vector Machine

- Support Vector Machine can be used for both classification or regression challenges.
- It performs classification by finding the hyperplane that maximizes the margin between the two classes with the help of support vectors.
- The distance between the support vectors and the hyperplane should be as far as possible.
- Support vectors are the extreme points in the datasets.
- We plot each data item as a point in n-dimensional space.
- Kernel converts non separable problem to separable problems by adding more dimension to it. It is most useful in non-linear separation problem. Kernel trick helps you to build a more accurate classifier.
    - Polynomial Kernel
    - Sigmoid Kernel
    - Radial Basis Function Kernel

Support Vector Machine Results			
	Gender	Age	Combine
Unigram	91.43	61.43	50.00
Bigram	85.71	59.14	45.71
Trigram	75.57	57.14	42.86

Table 6.2 Support Vector Machine results

6.3 Decision Tree

- The decision tree algorithm tries to solve the problem, by using tree representation.
- Decision Tree calculates the probability that a given record belong to each of the given category.
- Decision Tree splits the sample into two or more homogeneous sets based on most significant differentiator in input variables.
- Each internal node of the tree corresponds to an attribute or feature, and each leaf node corresponds to a class label.

Decision Tree Results			
	Gender	Age	Combine
Unigram	74.29	54.29	34.28
Bigram	64.29	54.29	35.86
Trigram	65.71	52.86	32.86

Table 6.3 Decision Tree results

6.4 Random Forest

- The random forest classifier is ensemble algorithm.
- Ensembled algorithms are those which combines more than one algorithms of same or different kind for classifying data.
- Random forest classifier creates a set of decision trees from randomly selected subset of training set.
- Then it aggregates the votes from different decision trees to decide the final class of the test data.
- It predicts based on the majority of votes from each of the decision trees made.
- Random Forest works well than decision tree because it reduces noise and aggregates votes of many decision trees for accurate results.

Random Forest Results			
	Gender	Age	Combine
Unigram	78.57	58.57	41.43
Bigram	72.85	57.14	37.14
Trigram	68.57	55.71	31.43

Table 6.2 Random Forest results

6.5 Experiment Analysis

Model	Gender			Age			Combine		
	Unigram	Bigram	Trigram	Unigram	Bigram	Trigram	Unigram	Bigram	Trigram
Support Vector machine	**91.43**	**85.71**	**78.57**	**61.43**	**59.14**	**57.14**	**50.00**	**45.71**	**42.86**
Naïve Bayes	80.00	65.71	60	57.14	50	45.71	34.28	27.14	27.14
Decision Tree	74.29	64.29	65.71	54.29	54.29	52.86	34.28	35.86	32.86
Random Forest	78.57	72.85	68.57	58.57	57.14	55.71	41.43	37.14	31.43

6.6 Conclusion

We presented our identification of gender and age-group in Multilingual Author Profiling on SMS on Roman Hindi and English language. Using the training dataset, we have developed the system using word based Term Frequency & Inverse Document Frequency (TFIDF) features and then classified with different ML classifiers i.e. Random Forest, Support Vector Machine, Naive Bayes, Decision Tree, and Logistic Regression. We have done the pre-processing by removing the stop words, non-ascii characters, and punctuations from the dataset. We have discussed the dataset descriptions and experiments used for the multilingual author profiling task. We have determined the results using Word Unigram, Bigram, and Trigram. We have tried till trigram approach as we observed that after trigram our accuracy got reduced it could be because of the dataset. From which for Gender Identification we got 91.43% accuracy by unigram with SVM classifier. We got 85.71% accuracy by bi-gram with SVM and 78.57% by trigram with SVM. For Age Identification we got 61.43% accuracy by unigram with SVM classifier. We got 59.43% accuracy by bi-gram with SVM and 57.14% by trigram with SVM and combine accuracy of 50%, 45.71%, and 42.86% with unigram, bigram, and trigram respectively.

6.7 Future Work

We aim to extend the model by making it more efficient by using different Techniques, we did not explore the deep neural network. Since our author profiles consist Roman Hindi, we have not tried language specific features to find out author profiles. So far the text contained two languages but in the future, it would be beneficial to include more South Asian languages as they are relatively less explored and contain potential to be very useful. We will collect more dataset. We would like to demonstrate other author traits like native language, native area, personality, type, qualification and occupation.

7. Bibliography and Citations

1. Fatima, M., Hasan, K., Anwar, S., Nawab, R.- M.- A. : Multilingual author profiling on Facebook. Information Processing & Management 53(4), 886-904 (2017).

2. Monika Briediene, Jurgita Kapociute Dzikiene. An Automatic author profiling from Non-Normative Lithuanbian Texts.

3. Fransisco Rangel, Paolo. On the impact of emotions on author profiling. Information Processing and Management 52(2016)73-92.

4. Murat Karabatak, Shannon Silessi, Cihan Varol: Identifying Gender from SMS Text Messages at 15th IEEE International Conference on Machine Learning and Applications (2015).

5. Jahna Otterbacher: Inferring Gender of Movie Reviewers at 19th ACM international conference.

6. K Santosh, Romil Bansal, Mihir Shekhar, and Vasudeva Varma: Author Profiling: Predicting Age and Gender from Blogs at CLEF 2013.

7. Jonathan Schler, Moshe Koppel, Shlomo Argamon, James Pennebaker : Effects of Age and Gender on Blogging at Conference of Computational Approaches to Analysing Weblogs – Stanford, California, USA – 2006.

8. Hernandez, D., Guzman-Cabrera, R., Reyes, A., Rocha, and M.-A.: Semantic-based Features for Author Profiling Identification: First insights. In: Proceedings of CLEF, (2013).

9. Francisco Rangel , Paolo Rosso .On the Identification of Emotions and Authors' Gender in Facebook Comments on the Basis of their Writing Style.http://www.uniweimar.de/medien/webis/research/events/pan-13/pan13-web/author-profiling.html.

10. Bayot, R., Gonçalves, T.: Multilingual author profiling using word embedding averages and svm. In: Software, Knowledge, Information Management & Applications (SKIMA), and 10th International Conference on. pp. 382–386. IEEE (2016).

11. Maharjan, S., Shrestha, P., Solorio, T.: A simple approach to author profiling in mapreduce. In: CLEF (Working Notes). pp. 1121–1128 (2014).

12. Argamon, S., Koppel, M., Fine, J. and Shimoni, A. R.: Gender, genre, and writing style in formal written texts. Text, 23, August 2003.

13. Ankush Khandelwal, Sahil Swami, Syed Sarfaraz Akhtar and Manish Shrivastava: Gender Prediction in English-Hindi Code-Mixed Social Media Content at Cornell University Library Arxiv (2018).

14. Schler, J., Koppel, M., Argamon, S., Pennebaker, J.: Effects of Age and Gender on Blogging. In American Association for Artificial Intelligence (2006).

15. Goswami, S.; Sarkar, S.; and Rustagi, M.: Stylometric analysis of bloggers' age and gender. In: International AAAI Conference on Weblogs and Social Media ICWSM (2009).

16. Nguyen, D., Gravel, R., Trieschnigg, D., Meder, T.: "How Old Do You Think I Am?" A Study of Language and Age in Twitter in 7th International AAAI Conference on Weblogs and Social Media ICWSM (2013).

17. Fransisco Rangel, Paolo: Use of Language and Author Profiling: Identification of Gender and Age in Natural Language Processing and Cognitive (2013).

18. T. Raghunadha Reddy, B. Vishnu Vardhan and P. Vijayapal Reddy: Author Profile Prediction using Pivoted Unique Term Normalization in Indian Journal of Science and Technology (2016).

19. Shlomo Argamon, Moshe Kopple, James W. Pennebaker, and Jonathan Schler: Automatically Profiling the Author of an Anonymous Text in Communication of ACM (2011).

20. Malcolm Corney, Alison Anderson, George Mohay, Olivier De Vel: Language and Gender Author Cohort Analysis of E-mail for Computer Forensics in Digital Forensic Research Conference (2017).

21. Lisa Kaati, Elias Lundeqvist, Amendra Shrestha and Maria Svensson: Author Profiling in the wild at IEEE Explore (2015).

22. https://www.guru99.com/nlp-tutorial.html

23. https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial

24. https://www.guru99.com/download-install-nltk.html

25. https://www.nltk.org

26. https://www.tutorialspoint.com/machine_learning_with_python