APPROVAL SHEET

Title of Thesis: Classification and Prediction of Newspaper Articles on the basis of Gender

Name of Candidate: Devisha Singh
Masters of Science, 2018

Thesis and Abstract Approved: 

Dr. Charles Nicholas
Computer Science

Date Approved: April 30, 2018

Devisha Singh

1138 Regina Drive Apt B Arbutus Maryland 21227| 240-441-2440 | devisha1@umbc.edu
LinkedIn: https://www.linkedin.com/in/devisha-singh-43097bbb/

EDUCATION

Master of Science in Computer Science University of Maryland Baltimore County GPA: 3.34/4 **2016-2018**
Bachelor of Technology in Information Technology from AK Garg Engineering College GPA 3.75/4 **2011-2015**

TECHNICAL SKILLS

Platform Windows
Programming Python, Matlab, R
Web languages JSP, HTML, CSS
Database MySQL, MS SQL Sever

Ø **Software** Microsoft Office, SAS, Eclipse, Wireshark
Ø **Applications** OpenCV, Tableau, Excel

PROJECTS

Gender Prediction and Classification of text on the basis of gender [Master's Thesis]

- Ø Created an XML parser in Python which parses the relevant text.
- Ø Developed a model using NLTK, Scikit-learn for python, used Logistic Regression for learning, and then compared the performance of different machine learning models and quantified the results.
- Ø Developed a classifier with high accuracy, precision and recall to predict the gender of the document.

Web Crawler

- Ø An application which crawled through the various websites to search for the most economical deals and discount offers for a particular product entered.
- Ø Built an application interface for the crawler, wrote scripts for web scraping using Object Oriented Design, UML modelling, dynamic data structures and then performed data wrangling, cleaning, transforming and merging the data frames.
- Ø Increased performance by implementing code migration to convert it to C++ using Cython.

Android Pay Protocol Analysis

- Ø NSA funded project to study and document the API of the android pay, the tokenization, card registration, user identification and in-store purchases mechanisms for the application.
- Ø Captured encrypted packets and analyzed these packet captures using Wireshark to identify transport layer protocols for message transmission and application level protocols for potential identification of API processes.
- Ø Analysis of vulnerabilities and potential risk involved with Packet duplication and Identity Theft.

WORK EXPERIENCE

Graduate Assistant University of Maryland Baltimore County **May 2017- Present**

- Ø Currently working as a Research Assistant on a project with the US Army on analysis of Physiological data related to stress for inducing adaptations to enhance regulation of brain states and cognitive task performance.
- Ø Primarily responsible for recording, collection and analysis of Electroencephalography(EEG) and Neuro-feedback data.

Graduate Research Assistant University of Maryland Baltimore County **May 2017-Aug 2017**

- Ø Responsible for analyzing OCO-2 satellite data provided by NASA related to Carbon Dioxide global measurements in UMBC's CHMPR lab.
- Ø Gather latitude and longitude points located in predetermined regions to calculate and compare the predicted CO2 fluxes.
- Ø Results employing a Feed Forward Backward Propagation Neural Network model on two architectures, an IBM Minsky Computer node and a hybrid version of the ARC D-Wave quantum annealing computer.

Project Intern at Reliance Communications Ltd, India **June 2014-Aug 2014**

- Ø Part of the project team for "Traffic Management & Implementation of GSM MSC Connectivity with RDN" focused on Reliance Development Network(RDN) Architecture up gradation to support WIN$-GSM project.
- Ø In CDMA project, I was responsible for managing and implementing, L2 extensions from MSC to BSC locations to support fast recovery of link failure.
- Ø Assisted in Maintenance of telecommunication rooms, onsite experience with handling of routers(CISCO).

PUBLICATIONS

"Comparative Study of Data Mining Algorithms through WEKA" **September 2015**
International Journal of Emerging Research in Management and Technology

ABSTRACT

Title of Thesis: CLASSIFICATION AND PREDICTION OF NEWSPAPER ARTICLES ON THE BASIS OF AUTHOR GENDER

Devisha Singh, Master of Science, 2018

Thesis directed by: Professor Charles Nicholas
Department of Computer Science

Categorizing text on the basis of author gender has been a long standing problem in the field of Machine Learning, taking gender as a basis for classification in different types of text. For the purpose of this thesis we focus on categorizing newspaper articles on the basis of gender, traditional machine learning techniques for classifying the text having been applied. Male and female writing styles have been identified.

The New York Times Annotated Corpus [18] licensed by Linguistic Data Consortium, containing approximately 1.8 million articles has been used. The article text is sorted, —articles containing definite male female author bylines and labels have been considered for classification and prediction initially, The text contains name of the author which has been matched against a male female labelled list to determine the gender of the author name. We try to predict the author of the authorless articles (containing articles written by collective boards such as editorials) on the basis of the model we built.

We also conduct a comparative study of different machine learning techniques like logistic Regression, Decision Tree Classifier, Support Vector machines and a few more to determine which learning method performs the best with the corpus.

CLASSIFICATION AND PREDICTION OF NEWSPAPER ARTICLES ON THE BASIS OF AUTHOR GENDER

by

Devisha Singh

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland Baltimore County in partial fulfillment
of the requirements for the degree of
Master of Science
2018

Advisory Committee:
Professor Charles Nicholas, Advisor
Professor Frank Ferraro
Professor Samuel Lomonaco

Dedication

To my brother and parents.

Acknowledgments

I owe my success in completing the thesis to everyone who has made this thesis a possibility, with their experience and support this would not have been possible

First and foremost I would like to thank my advisor, Professor Charles Nicholas for allowing me to work under his guidance. With his experience he guided me through the nuances of the thesis easily. I am grateful to him to be my mentor and guide, helping me with the challenges I faced during the course of my thesis. He was always appreciative and supportive of my ideas, provided that invaluable touch of his experience and then there were times when I was not sure of what I was doing, he motivated me a showed me the right way, I learned a lot while doing my thesis. It was an honour to be able to work under him.

I would also like to thank Dr Frank Ferraro. Without his extraordinary skills and computational expertise, this thesis would not have been a reality. He is a specialist in machine learning and natural language processing and provided me with small tricks and tips on how to proceed with the thesis.

I owe my deepest thanks to my family- whose unending support through the ups and downs of the thesis have been of immense help. I cannot express in words the gratitude I have for them.

I would also like to thank Almighty! Trust and Belief make you go a long way.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SVC	Support Vector Classification
SVM	Support Vector Machine
TF	Term Frequency
IDF	Inverse Document Frequency
BOW	Bag of Words

Chapter 1: Introduction

1.1 Machine Learning

Machine Learning is a domain in computer science which relates to the capability of computer machines to learn from examples and make decisions based on previous observations and reactions etc.

Informally, we can say that Machine learning is a science of getting computers to learn to perform multiple tasks or functions without being explicitly programmed to do so.

Machine Learning is being used in a variety of day to day applications such as tasks, search engines, analyzing and predicting stock markets, pattern recognition such as facial recognition, speech recognition, image recognition. Its widely used in forensics for DNA analysis for diagnosing medical problems. All of these applications involve the computer either learning from the data which has been given or learning from a defined pattern observed in the data. The areas where machine learning technology is being used is increasing as time progresses.

A more formal definition of machine learning as provided by Tom Mitchell: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured

by P, improves with experience E."

In context to our thesis work, it can be aptly said that,

E = the experience of classifying numerous articles as male or female

T = the task of classification.

P = the probability that the program will classify accurately. [15, 20]

1.2 Document Classification

One of the problems being addressed by Machine Learning is Document Classification, It has been a long standing problems in this domain, with several techniques and machine learning algorithms applied to it. Documents can be classified on the basis of their type, for example image, text, audio.

Taking into account Text classification, it's about categorizing or classifying text on the basis of various features like genre, type, author. For the purpose of text classification various machine learning techniques have been used to get results. A lot of research has been done in correctly classifying and predicting different type of texts, for example email blogs, posts, novels, formal legal documents. [2, 4]

Our focus would be categorizing documents, in this case text articles on the basis of gender. We aim to use Statistical text categorization technique called the bag of words and few other approaches to classify the text as either male or female.

1.3 Bag of words

In order for the computer to understand document classification we use a technique which is called the Bag of words which specifies that each document, in this case each newspaper article, is represented mathematically as a vector $x = (0, 1)^n$ where n represents dictionary of words present in a set of documents, when a word corresponding to index i is present in the document then $x_i = 1$,if it is not present $x_i = 0$.

To further elaborate let $(d_1, d_2...d_n)$ be the words in a particular document. We represent each document as $x = (0, 1)^k$, we have a dictionary of words say $(w_1, w_2...w_n)$. So for any word d_i occurring in the document the vector $x_i = 1$ iff $w_i = d_i$ otherwise $x_i = 0$, that is, a word in the dictionary matches a word in the document. [15]

1.4 Term Frequency - Inverse Document Frequency

Considering a classic vector space model for a any dimension d, where each dimension is different , each word d_i in a document is a feature. We define Term Frequency as the number of times a term is occurring in a document. Since the size of the document varies, the term frequency can be calculated as

$$TF_t = \frac{n_{terms}}{Total_{terms}} \tag{1.1}$$

In the equation (1.1), $Number_t$ represents the number of times a particular

term t appears in the document and $Total_{terms}$ represents the total number of terms in the document. Whereas the inverse document frequency tells us the importance of that particular term in a document and is represented by the logarithm of total number of documents divided by the number of documents containing the term. It can be written in the form of an equation as

$$IDF_t = log\Big(\frac{total\, number\, of\, documents}{number\, of\, doucments_{term}}\Big) \tag{1.2}$$

After defining the term frequency and the inverese document frequency, a weighing scheme based on the tf-idf form is defined as

$$tf - idf = TF * IDF \tag{1.3}$$

This equation (1.3) multiplies and combines term frequency and inverse document frequency, which results in a composite weight for each term in each document. The $tf - idf$ weighting methodology assigns to a term say t a weight in document say d. The equation (1.3) can be further written as

$$\text{tf-idf}_{t,d} = \text{TF}_{t,d} \times \text{IDF}_t. \tag{1.4}$$

In other words, $\text{tf-idf}_{t,d}$ assigns to term t a weight in document d that can have three possibilities: Firstly, $tf - idf$ is maximum when t appears many times in a small number of documents (these documents have high discriminating power); Secondly, $tf - idf$ weight is lower when the term occurs fewer times in a document, or occurs in many documents(thus offering less significance); Lastly, $tf - idf$ weight is the lowest when the term appears in almost every document.

So summarizing we can aptly say our aim is to develop a model which could accurately classify male and female by analyzing the text document in this case the article and we aim to use that learning model to predict the gender or attribute gender to the author-less articles such as articles written by editorials. The question lies whether a model built can be used for other relevant text classification of similar articles.

The thesis first covers some previous and similar work which has been done in this area, work mainly related to categorizing, and classifying various corpora on the basis of gender, the corpus ranges from novels to web blogs to tweets. These works give an insight into the writing styles which are unique to a particular gender. We then mention how our work is different from the work previously done. Then a brief description of the theoretical concepts of the various models have been covered. These models serve as a basis for deciding the final learning model, a comparative analysis of various models having been done on a sample of the corpus performing a 10 fold cross-validation to decide the best fitting model which has been used on the whole corpus finally.

The next few chapters go into the detail of how the data is extracted and processed to be given as an input to our learning model. We use a 90 percent 10 per cent train/test stratified split. The trained model is finally used to predict the gender of the author-less articles. Author-less articles imply the articles which have been authored anonymously or by collective boards.

The last section of the document aims to answer the research question which was proposed initially, aims to draw certain conclusions which were observed from

the process of learning, training testing and prediction. At the end, we try to draw out a comparison, a parallel between the percentage of journalists and the population of the United States to see if a gender-bias exists when it comes to the journalism. We also try to expand the scope of this research into a possible four-way and a six-way classification which could lead to interesting results.

Chapter 2: Previous Work

2.1 Classification of Gender on the Basis of Genre

Kopell et al [1] used lexical and syntactic features for the purpose of categorizing the text based on gender. The authors used function words and Part of Speech (POS) as features for categorizing and determining the author of novel based corpus The corpus used was genre specific and consisted of 566 documents from the British National Corpus.

They defined a weight vector w, for each training document and classified it as a male or a female on the basis of a pre-determined threshold. To determine the weight vector they used was modification of the Exponent Gradient (EG) algorithm (Kivinen et al) [16] which is a generalized form of Balanced Window.

In mathematical terms it can be theoretically said that

$$w.x > T \quad iff\ x\ is\ authored\ by\ a\ female \tag{2.1}$$

That is to say, if the vector dot product of w and x increased above a certain threshold if the document was authored by a female otherwise it is classified as male.The authors performed 56 fold cross-validation with 10 examples in each fold

using 3 feature sets which consisted of firstly, function words, secondly parts-of-speech only and third was a combination of both function words and parts-of-speech. For function words, the accuracy was 73.7 per cent of the documents which were correctly classified, for parts-of-speech 70.5 percent, and for the full feature set, the accuracy improved to 77.3 percent with approximately 1 percent standard errors. They concluded that using a combination of both Function words and Parts of Speech yielded the best results. [1,3]

They concluded that a better result was possible on using a combination as it could exploit unique anomalies such typical verbs or prepositions used distinctly by males or females, they made some interesting observations in relation to the writing styles.

2.2 Predicting gender for Blog posts

In this work the authors Zhang et al tried something similar to what we're trying to do. They used data from various blog posts to predict the gender of the blog posts. They defined two sets of gender classes male and female, and selected features using various statistical modelling techniques like Information Gain, mutual information, and performed comparative analysis of the of different classification algorithms. They obtained a prediction accuracy of 72 percent using Information Gain (IG) for feature selection criterion, and SVM as the classifier. They also observed that features slightly improves prediction accuracy in combination with feature selection mechanisms. [5]

In another similar work, the authors collected data blog posts from many commonly used blogs and blog search engines.The data set consisted of 3100 blogs. Each blog was labeled with the gender of its author. The gender of the author was determined by visiting the profile of the authors, Mukherjee et al proposed a novel class of features which were Part of Speech sequence patterns that are able to capture complex stylistic similarities of both male and female authors. [2]

2.3 Classification on Twitter Data by n gram analysis

In this specific gender classification on real-time Twitter data they tried to identify genders of the users on Twitter by representing individual tweets as a vector based on 1 through 5-gram features. For the graphical features, like emoticons and the misspelled words or slang expressions n-grams were employed instead of traditional dictionaries. To take out and select the describing features or features which give us some information and improve the classification accuracy as well as the run-time of these algorithms, 6 feature selection algorithms were used Results were used to select the best ranked features. The authors demonstrate the effectiveness of the features selected after using the selection algorithms . Algorithms which were used are Prceptrons, which is a simple neural network approach, and the Naïve Bayes approach which is a probabilistic model. Perceptron preformed relatively well with very high precision 97 percent, and a balanced the accuracy of 94 percent which was outperformed by Naïve Bayes scoring between 90 percent and 100 percent for all metrics. The performance of the Perceptron and the Naïve Bayes stream algorithms

for gender identification of Twitter users demonstrate the value of the n-gram feature representations as well as the feature selection techniques. [4]

Chapter 3: Machine Learning

3.1 Introduction

Machine learning can be achieved by two ways: it can be supervised or an unsupervised. In supervised learning, the system receives a data-set with various parameters and decisions/classification, from which it derives a mathematical function, which automatically maps an input signal to an output signal. This research has made the use of supervised methods. Unsupervised learning, on the other hand, means that the machine learns from its own mistakes and corrections from the moves it makes and draws conclusions as a consequence of its actions, without previously referring any of the predetermined observations. It can be said to be learning by trial-and-error. Compared to supervised learning, unsupervised methods have a low performance at the start, they need more tuning, but as time progresses, they tune themselves, and performance increases. It can be said that using unsupervised learning, a classifying system should be able to set up a hypothesis that no human can conclude, due to the complexity. If unsupervised methods were used for this project, the machine learning system would have to find out the learner stage hypothesis all on its own, which would probably require much more training data than is available. One would run the risk of concluding a hypothesis too complex or specific.

To quantify classifier performance given by a machine learning model, either a special testing set or a cross-validation technique may be employed. A test set contains pre-classified data, but this is different from the training set and is used only for evaluation, not for training. If we have less data, it is advisable to use cross-validation in order not to waste any data. We discuss k-fold cross-validation in detail in the later chapters. Cross-validation could be useful to enhance classifier results and its performance on the dataset; all data are used both for training the classifier and for testing its performance.

More training data example does not necessarily imply better performance. Even though the classifier becomes better on the training set it could actually perform worse on the test data. This is possible due to the overfitting of the classifier, it can fit tightly to the training data and the border between classes would be jagged rather than smooth, unlike how it usually should be. [10]

3.2 Logistic Regression

In logistic regression we have a family of functions h from R_d to the interval $[0, 1]$.Logistic regression is used for classification tasks: We can interpret $h(x)$ as the probability that the label of x is 1. The hypothesis class associated with logistic regression is the composition of a sigmoid function $\sigma : R[0, 1]$ over the class of linear functions L_d. In particular, the sigmoid function used in logistic regression is the logistic function, defined diagrammatically as shown in Figure 3.1:

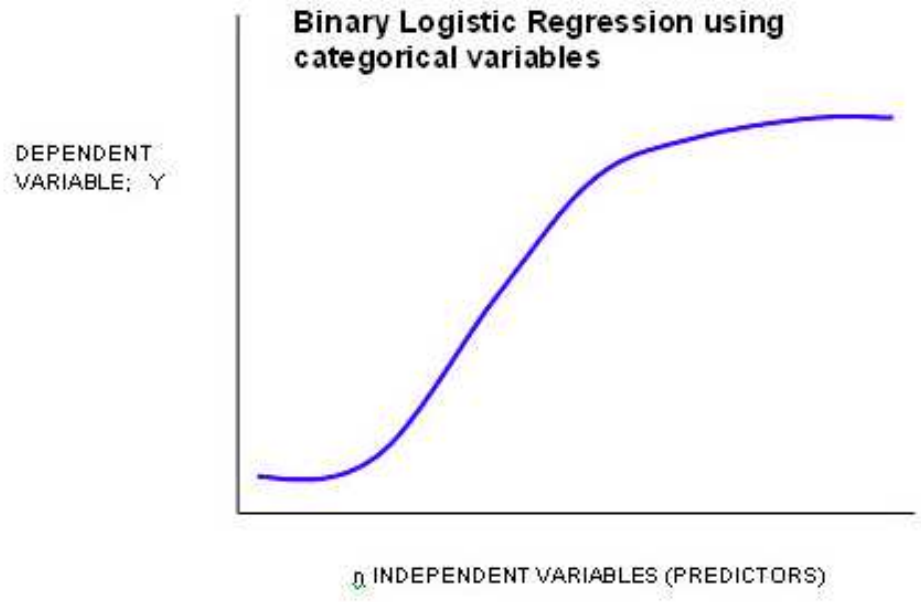

Figure 3.1: Logistic Regression.

3.3 Naive Bayes

Naive Bayes is the second approach we tried. Let $C = (g1, g2)$ be the gender class, and $F = (f1, f2, ...fn)$ be the features of the class, So Bayes theorem says that [8]:

$$\frac{P(g|F) = P(g)P(F|g)}{P(F)} \tag{3.1}$$

The naive Bayes assumption is that:

$$P(F|g) = \prod_{i=1}^{n}(fi|g) \tag{3.2}$$

So the clasification algorithm is:

$$argmaxP(C = c|F) = argmaxP(C = c)P(f = fi|C = c) \tag{3.3}$$

3.4 Decision Tree Learning

A decision tree is a tree structured classifier or a leaning model, which consists of rules which leads to a decision. Each rule may point or expand to another rule or to another decision. For example, you could say that you like playing Football, unless the weather is inclement, and diagrammatically represent the unsuitable conditions in a tree, such as the one below.

Using machine learning algorithms, a computer can infer such decision trees from tables with examples showing various conditions (attributes) and outcomes or classifications. shown as:

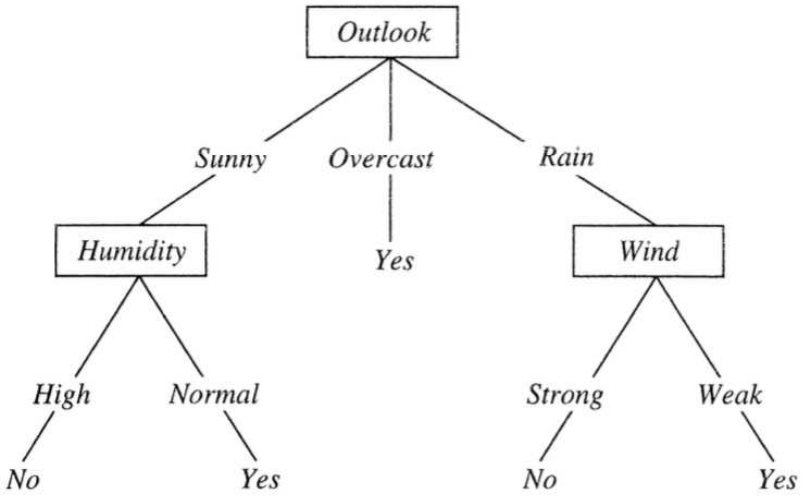

Figure 3.2: Decision Tree Learning

.

3.5 Random Forest Classification

[19] Random Forests is also a tree based classifier and it grows many classification trees. For the purpose of classification of a brand new object from an input vector, the input vector is written down each of the trees in the forest. Every tree gives us a new classification, and it can be aptly said that the tree casts its vote for that class. The forest at the end decides and chooses the classification getting the most votes (over all the trees in the forest).

Each tree is grown as follows:

Considering any number of cases in the training set, let the set have random number of cases say N, The classifier samples N cases at random - but with replacement, from the original data. This sample obtained as a result will be used as a training set for growing of the tree.

If there are some input variables say H, a number $h << H$is defined and written such that at each node, h variables are picked out at random out of the H variables and the best split among these h is used to split the node. The value of h is taken to be constant during growing of the forest. Each tree is grown to the largest extent possible. There is no pruning.

The Forest error rate depends on two things: The correlation between any two trees in the forest. If we increase the correlation it increases the error rate.

Reducing the size of h decreases the correlation and ultimately it reduces the

trees strength. Enhancing it increases both of them. Somewhere in between is an "optimal" range of m - usually quite wide. [13]

3.6 Bernoulli's Naive Bayes

In this model, when we are calculating the probability of a document, we multiply the probability of all the attribute values, which includes the probability words that have no occurrence in the document. We interpret the document to be the "event," and the occurrence and the non occurrence of words to be attributes of the event. This is the description of the distribution based on a multi-variate Bernoulli event model. This methodology is more traditional to the Bayesian networks, and is appropriate for tasks that have a fixed number of attributes. The approach has been used for text classification in most cases.

We are making the assumption that text documents are generated by a combination of models parameters by θ. The resulting model consists of components c_j such that $C = (c1, ..., c|C|)$. Each component has a disjoint subset of θ. Thus we can say that a document d_i, is created by Firstly, selecting a component according to the priors, P(c_j —θ), then secondly the mixture component generate a document according to its own parameters, with distribution P($d_i|c_j$; θ). The likelihood can be written as:

$$P(d_i|\theta) = \sum_{j=1}^{C}(P(c_j|\theta) + (1P(d_i|c_j;\theta) \tag{3.4}$$

Each document has a class label. We assume that there is a one-to-one cor-

respondence between classes and mixture model components, and thus use c_j to indicate both the jth mixture component and the jth class.

In the multivariate Bernoulli event model, a document is a considered to be a binary vector over the space of words. Given random vocabulary of words V , and a dimension of the space S, $S1, ..., |V|$, corresponds to a word w in the vocabulary. Dimension S of the vector for document d_i is written , and is either 0 or 1, indicating whether word w is present in the document, it determines its occurrence at least once in the document. With such a document representation, the following assumption has been made : that the probability of each word occurring in a document is independent of the occurrence of other words in a document. Then, the probability of a document given its class is simply the product of the probability of the attribute values over all word attributes:

$$P(F|g) = \prod_{t=1}^{V} (B(w_t|c_j; \Theta) + (1B_{it})(1P(w_t|c_j; \theta) \tag{3.5}$$

Thus given a generating component, a document can be seen as a collection of multiple independent Bernoulli instances, one for each word in the vocabulary, with the probabilities for each of these word events defined by each component, $P(wt|cj; 0)$. This is equivalent to viewing the distribution of documents as being described by a Bayesian network, where the absence or presence of each word is dependent only on the class of the document.

3.7 Multinomial Naive Bayes

The multinomial model captures word frequency information in documents. Considering a New York Times Annotated Corpus article, since every news article is dated, thus has a number, the number token in the multi-variate Bernoulli event model is uninformative.

In the multinomial model, a document is an ordered sequence of word events, drawn from the same vocabulary V . We assume that the lengths of all the documents is independent of its class. The second assumption is that naive Bayes assumption: that the probability of each word event in a document is independent of the context of the word and its given position in the document. In conclusion we can say that each document d_i is taken from a multinomial distribution of words with multiple independent trials which is equal to the length of d_i. [13]

3.8 Selecting a model

For deciding a model we run it on a sample of training data and obtain cross-validation scores, The package SKlearn has been used to obtain the performance of these models on the sample training set, Figure 3.3 shows pseudo code for the various model which have been used for choosing and deciding the model. Note that in order to use the Sophisticated Gradient Descent the parameters which have been used for better tuning is loss='hinge', penalty='l2', alpha=1e-3 randomstate=42

A k-fold cross validation is often used for model selection (or parameter tun-

```
def apply_classification_using_models():
    print("Loading in the data set")
    M = Model(df=pd.read_csv('/Users/Devisha/Desktop/ArticleClassifier/Resources/author_article_gender.csv'))
    print("data has been loaded")
    # predict_proba() # < ---
    # Below is the classifiers list.
    classifier_list = {"logistic": LogisticRegression(solver='newton-cg')  # ,
                       "Decision Tree Classifier :": DecisionTreeClassifier(),
                       "Linear SVC": LinearSVC()  # ,
                       "svc classification": SVC(kernel='rbf'),
                       'One vs one classifier': OneVsOneClassifier(estimator=1),
                       "multi Naive ": MultinomialNB(),
                       "bernouli naive bayes": BernoulliNB(),
                       "Ada boost classifier": AdaBoostClassifier(),
                       "sophisticated gradient decent": SGDClassifier(loss='hinge', penalty='l2', alpha=1e-3,
                                                                      random_state=42),
                       "Gaussian bayes classifier :": GaussianProcessClassifier(),
                       "KNN classifier": KNeighborsClassifier(n_neighbors=10),
                       "Random Forest": RandomForestClassifier()
                       }
```

Figure 3.3: Pseudocode.

ing), and once the best parameter is chosen, the algorithm is retrained using this parameter on the entire training set. A description of k-fold cross validation for model selection is given in the following. The procedure receives as input a training set, S, a set of possible parameter values, , an integer, k, representing the number of folds, and a learning algorithm, L, which receives as input a training set as well as a parameter . It outputs the best parameter as well as the hypothesis trained by this parameter on the entire training set. [10]

The cross validation method often works very well in practice. The training set consist of approximately 6,000 documents. X is considered as the input parameter and y is considered as the output parameter, and the different models evaluated are used in place of Learning algorithm L. Input of possible parameter values is received and a 10-fold cross validation is performed. We will perform a 10 fold cross validation on the whole data set to determine the model which suits best, the whole training set is used for training the models and a 10-way cross validation is per-

formed and accuracy scores, average cross validation scores are calculated in order to determine the results.

Classifier	Accuracy	Cross Validation Score
Support Vector Classification	53.9	57
Decision Tree	57.21	58.17
Random Forest	61.2	57.1
Logistic Regression	68.75	65.0
Gradient Descent	63.7	60.24
ADA Boos Classifier	58.88	58.02
Multimomial Naive Bayes	63.94	59.9

Table 3.1: Cross Validation Scores

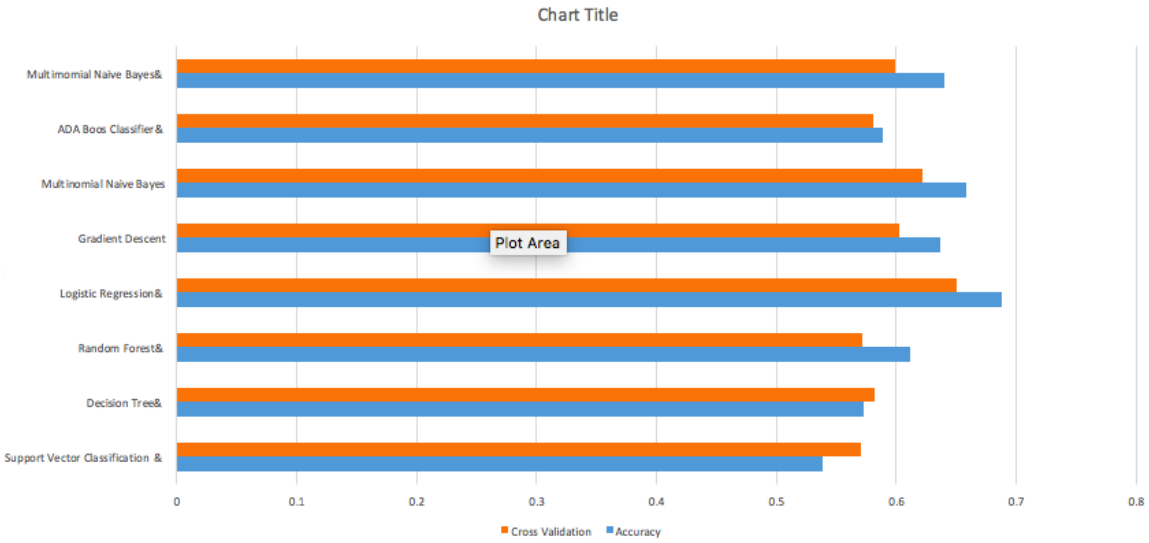

Figure 3.4: Comparison of Cross Validation Scores.

After performing the 10 way cross validation, we see that Logisitc Regression gives us an accuracy of 63.7 percent on a sample of data and Multinomial Naive Bayes also performs well on the sample with an accuracy of 63.94 per cent, Multinomial is said to do well in text based classification and as expected its performance is better than most of the classifiers. Decision tree and Random forests give us a lower accuracy and cross validation scores. Logistic Regression has the highest accuracy of 68.75 per cent and a cross validation score of 65 per cent. We choose Logistic Regression after parameter tuning and select it as a model for training and testing.

Chapter 4: Pre-processing

As explained in the previous chapters we would use Logistic Regression, since it performs the best on this data set. The dataset being used is in the XML documents which conforms to the News Industry Text Format(NITF) Specification.

The format consists of many tags but the tags considered for the purpose of data collection in this study are the bylines which specifies the byline of the article as it appears in the article, usually signed by the author name, The articles not containing the byline are unsigned, meaning they have no author and generally consist of editorials or other articles. The body content consists of the lead paragraph and the full body text which are used.

The corpus contains several metadata annotations, As mentioned above some of them don't contain author name and there are different articles like paid death notices, advertisements, announcements etc which has not been considered for this study, There are several annotations like publication date, different sections of the articles, which can be used for future scopes. A parser parses through the XML document to record the byline, and extracts the article paragraphs from the document and dump it into an obj file. Solo authors male/female have only been considered. The parser uses packages like Beautiful Soup4 (commonly called bs4) for parsing,

and pickle for storing and loading of objects from the file.

The parser ignores the documents that do not contain a byline and skips the advertisements and paid death notices and creates an object file for all the articles that contain an author. It denotes the statistics of all the files which have been parsed and which have been ignored. Here is a sample of what the data looks like after being parsed by the parser and before being used for training.

```
In [6]: df.head()
```

Out[6]:

	Unnamed: 0	author	article	gender
0	0	bob tedeschi	Smilebox.com offers customers ability to build...	m
1	1	sharon waxman	Judith Regan says she is preparing to file law...	f
2	2	kathryn shattuck	11 A.M. (HGTV) ROSE PARADE 2007 -- HGTV presen...	f
3	3	nate chinen	Nate Chinen reviews performance by Gov't Mule ...	m
4	4	julie galambush	Julie Galambush reviews book The Misunderstood...	f

Figure 4.1: Parsed Data

4.1 Processing

The XML parser creates the an object file of the sample data which is shown above, which is then used to generate data for the machine learning model by a second script. The Script reads and loads the file containing the common names and their labels, and creates an object file. Both the tagged and the un-tagged object files are loaded with the help of pickle.

The byline containing the author name is matched with a list of common names obtained from the Social Security Administration which consists of the male

and female labels and this scripts assigns labels to the data. If the name in the byline is matched with one of the common names in the list of common names then that particular document/article is tagged as a male or a female. At the end all the documents containing definite labels are collected in a CSV file.

Given below is a diagrammatic Representation of the Data which is Raw, undergoes processing and then training and finally the trained model is used for prediction of the unattributed articles.

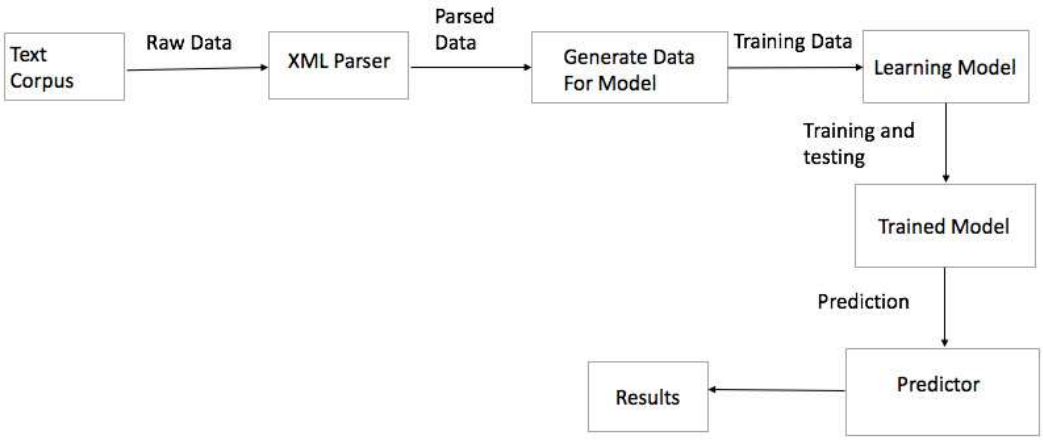

Figure 4.2: Flow of data.

4.2 Creating a pipeline

The text must be parsed to remove words. Then the words need to be encoded as integers or floating point values for use as input to a machine learning algorithm, called feature extraction (or vectorization).

The scikit-learn library offers easy-to-use tools to perform both tokenization and feature extraction of your text data.

The CountVectorizer provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary.

On implementing a count vectorizer. We get a coded vector with a length of the entire vocabulary and a count for the number of times that each word has appeared in the document.

Here is a pseudo code used to created the pipeline.

```
def set_classifier_pipeline(self, classifier):
    self.pipeline = Pipeline(
        [
            ('vectorizer', CountVectorizer(stop_words='english', max_df=0.8)),  # using the english stop words
            ('tfidf', TfidfTransformer()),  # tf idf transformer
            ('classifier', classifier)  # classifier any !
            # min 60 , max 66 , avg 64
        ]
    )  # A pipeline for the logistic Regression with the Multinomial , newtons solver.

    return

# getting the pipeline
def get_pipeline(self):
    # Logistic Classification multi nomial
    self.pipeline = Pipeline(
        [
            ('vectorizer', CountVectorizer(stop_words='english', max_df=0.8)),  # using the english stop words
            ('tfidf', TfidfTransformer()),  # tf idf transformer
            ('classifier', LogisticRegression(solver='newton-cg', multi_class='multinomial'))  # multiclass based
            # min 60 , max 66 , avg 64
        ]
    )  # A pipeline for the logistic Regression with the Multinomial , newtons solver.
    return self.pipeline  # Pipeline.
```

Figure 4.3: Pipeline using Count Vectorizer and Tf-IDF Transformer.

Because these vectors will contain a lot of zeros, we call them sparse. Python provides an efficient way of handling sparse vectors in the scipy.sparse package.

Also in the pipeline we use the TF IDF format which has been described in detail before.

The TfidfVectorizer tokenizes the documents, learns the vocabulary present and the inverse document frequency weightings, and also allows us to encode new documents. The TFIDF Tranformer is used with CountVectorizer for our model, to calculate the inverse document frequencies and start encoding documents.

The classifier with maximum accuracy scores in this case, Logistic Regression, is used in the pipeline.

4.3 Learning Model

We perform a two way binary classification on the final processed data set using Logistic Regression, performing a 0 1 classification, namely classify male as 0 and female as 1. The default threshold for the model is 65 percent, and the data is split into a 90, 10 per cent stratified split.

The model calculates the accuracy, precision and recall scores for logistic regression. Logistic Regression models are generally fitted on maximum likelihood.

We discuss in depth the two-class case, since the algorithms simplify considerably. The log likelihood of N observations can be written as:

$$l(\theta) = \sum_{i=1}^{N} logp_{gi}(x_i; \theta)$$

It is more convinient to code the two-class via a 0/1 response y_i, where $y_i = 1$ and $y_i = 0$ when $g_i = 2$. Let p1 (x;) = p(x;) and p2(x;) = 1 p(x;).The log-likelihood can be written as

$$l(\beta) = \sum_{i=1}^{N} \{y_i log p_{gi}(x_i;\beta) + (1-y_i) log(1-p(x_i;\beta)\} = \sum_{i=1}^{N} \{(y_i;\beta^T x^i - log(1+e^{\beta^T x^i})$$

It seems that $\beta = 0$ is a good starting value for the iterative procedure, although convergence is never guaranteed. Typically the algorithm does converge, since the log-likelihood is concave, but overshooting can occur. In the rare cases that the log-likelihood decreases, step size halving will guarantee convergence.

We use the Newton method/ newton solver in logistic regression for our dataset, Newton's Method is an iterative equation solver: it is an algorithm to find the roots of a polynomial function.

In the simple, one-variable case, Newton's Method is implemented as follows, For input parameter's x and y, the pseudocode for Newton's Method:

Here is the approach considering x as the input pararmeter and y as the output parameter

Find the tangent line to f(x) at point (x_n, y_n)

$y = f(x_n)(xx_n) + f(x_n)$

Find the x-intercept of the tangent line, $x_n + 1$

$0 = f(x_n)(x_{n+1}x_n) + f(x_n)$

$f(x_n) = f(x_n)(x_{n+1}x_n)$

$x_n + 1 = x_n f(x_n) f(x_n)$

Find the y value at the x-intercept. $y_{n+1} = f(x_{n+1})$

$If y_n + 1 y_n 0$:

return $y_(n+1)$ because we've converged!

Else update point (x_n, y_n), and iterate

$x = x_n + 1, y = y_{n+1}$, goto the first step.

Chapter 5: Quantitative Results

5.1 Processing

The model parsed approximately 1.8 million documents. Approximately 500,000 documents article consisted of solo male and solo females, The Logistic Regression when used to classify the dataset, gives us accuracy of 85.1 per cent. The precision, recall and f1 scores for the male and the female labels for a sample of dataset are given below in the table:

Name	Female	Male
Precsion	91.6	85.2
Recall	36.6	99.09
F1	52.3	91.6
Support	30	111
Accuracy	85.81	85.81

Table 5.1: Scores for Default Logistic Regression Learning model

It can be seen clearly from the table Male authors have high recall score of 99 per cent while female have recall score of 52.3 per cent, indicating the less percentage

Name	Male	Female
Precsion	82.78	82.82
Recall	92.2	65.9
F1	87.21	73.94
Support	31977	18140
Accuracy	82.78	82.78

Table 5.2: Scores for Tuned Logistic Regression Learning model

of female authors. Logistic Regression models have better precision scores for both the males and the female with 91 per cent and 85.2 percent respectively. Note that these score are obtained in Tale 5.1 on a sample of data set of 200 documents with default threshold set to 50 per cent confidence, we observe that the when we use logistic regression on the whole data set (Table 5.2) we obtain better recall scores of 92.2 percent and 65.9 percent which is better than what we obtain on a default threshold. Another observation is the accuracy decreases slightly from 85.8 percent to 82.78 percent when the length of data set increases. One of the things which can be concluded is that we are able to increase the Recall scores for the female labels thereby increasing the f1 measure of the same, this is done by selecting the threshold which is a trade off which aims to increase the overall performance of the classifier:

In figure 5.1 we present the confusion matrix obtained for the whole data set

11972 (True Positive- TP)	6168 (False Positive -FP)
2483 (False Negative- FN)	29494 (True Negative -TN)

Figure 5.1: Confusion Matrix

5.2 Prediction Results

The Logistic Regression model is now used to predict the articles which were collected in a obj file which consist of collective boards in the byline, note that these include articles for Editorial Boards, National Board, Sports Board and Arts and Science Board.

The Figure 5.2 below shows the percentage of articles predicted as male and percentage of articles predicted as female by the model.

Predicted number of articles 330,004 Classified as male 262,789 Classified as female 67215 Not classified due to less confidence: 214813.

5.3 Statistics

As the statistics show the New York Times print industry has been dominated by male journalists and this study conducted confirms the same. Here are some statistics showing the comparison of various print media and their male and female. Figure 5.3 shows that the New York Times consisting of 2454 male journalists and

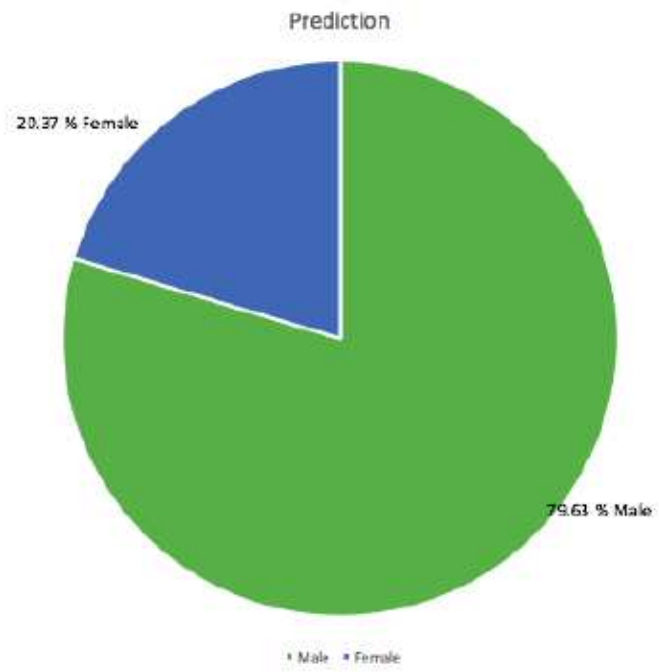

Figure 5.2: Prediction Results

1102 female journalists has the widest gender gap while Chicago Sun Times has the smallest, but men outnumbered women at all the 10 newspapers. (American Society of News Editors) [17]

It is clear that New York Times has the widest gender gap, having 2454 males to 1102 females, while Chicago Times shows the least gender bias. The bias seems to be proven in the study since the corpus has male authored articles around 80 per cent and female authored articles around 20 per cent. [17].

The model developed by us predicts 79.6 percent male, 20.37 percent female. We have no ground truth to determine the accuracy of the model but the percentage

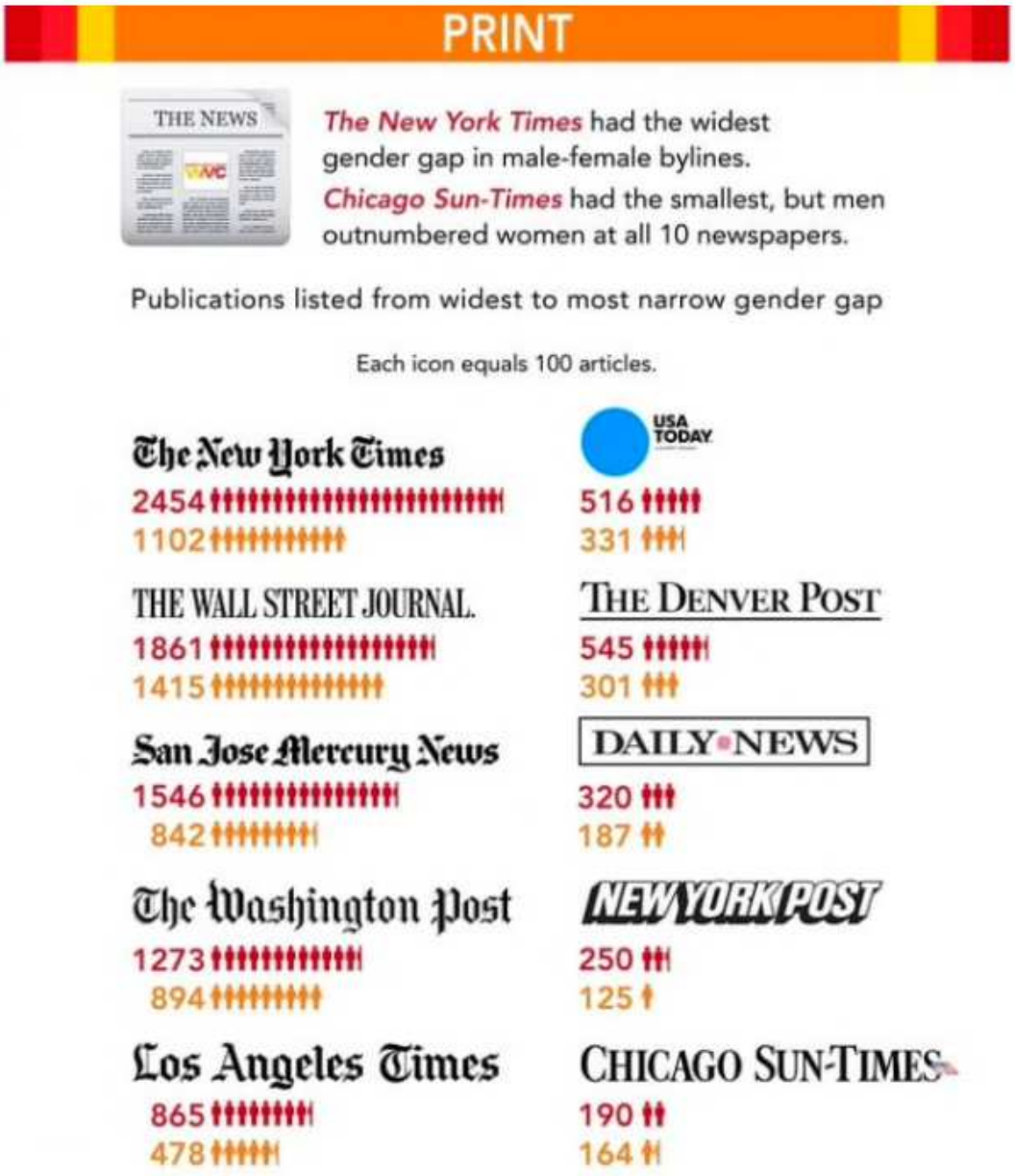

Figure 5.3: Comparative Analysis of the Gender distribution.

of predictions is consistent with the NYT demographics as mentioned in Figure 5.4 and 5.5.

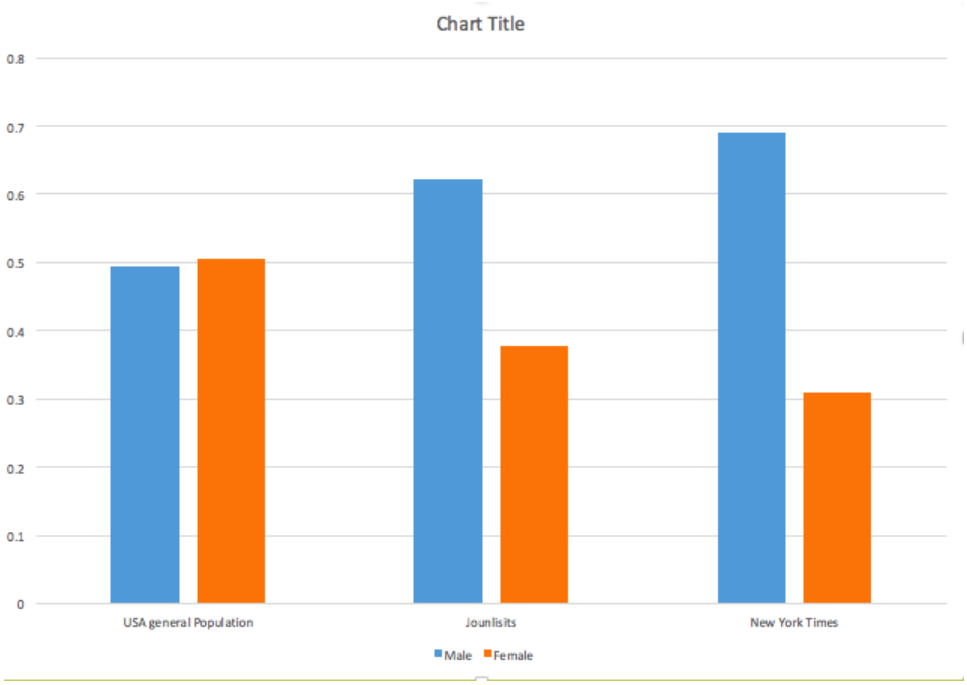

Figure 5.4: Gender Bias.

	Supervisors				Copy/Layout Editors Online Producers				Reporters/Writers				Photographers/Artists/ Videographers				TOTAL			
	Men		Women		Men		Women		Men		Women		Men		Women		Men		Women	
	No.	Pct.	No.	Pct.	No.	Pct.	No.	Pct.	No.	Pct.	No.	Pct.	No.	Pct.	No.	Pct.	No.	Pct.	No.	Pct.
2013	5,944	65.4	3,143	34.6	4,794	60.1	3,186	39.9	10,842	62.2	6,580	37.8	2,622	75.1	871	24.9	24,202	63.7	13,780	36.3
2012	6,619	65.8	3,447	34.2	5,114	57.7	3,752	42.3	11,210	62.0	6,881	38.0	2,651	74.8	892	25.2	25,595	63.1	14,971	36.9
2011	7,081	65.4	3,746	34.6	4,677	58.5	3,319	41.5	11,348	61.3	7,175	38.7	3,143	73.7	1,120	26.3	26,248	63.1	15,360	36.9
2010	7,037	66.4	3,560	33.6	4,668	58.0	3,385	42.0	11,537	61.9	7,115	38.1	3,036	73.0	1,120	27.0	26,278	63.4	15,180	36.6
2009	7,983	65.2	4,251	34.8	5,445	58.2	3,906	41.8	12,377	61.3	7,813	38.7	3,578	73.1	1,317	26.9	29,383	63.0	17,287	37.0
2008	8,796	64.8	4,776	35.2	6,208	58.2	4,456	41.8	13,886	60.9	8,911	39.1	4,058	72.9	1,508	27.1	32,947	62.6	19,651	37.4
2007	9,049	65.4	4,792	34.6	6,470	58.0	4,689	42.0	14,653	60.3	9,633	39.7	4,186	72.7	1,573	27.3	34,358	62.4	20,687	37.6
2006	8,541	64.4	4,716	35.6	6,094	58.5	4,331	41.5	14,566	60.3	9,604	39.7	4,144	72.6	1,566	27.4	33,345	62.3	20,217	37.7
2005	8,564	65.2	4,580	34.8	6,172	58.8	4,320	41.2	14,798	60.1	9,807	39.9	4,281	72.6	1,613	27.4	33,814	62.5	20,320	37.5
2004	8,583	65.8	4,471	34.2	6,188	58.6	4,366	41.4	14,994	60.4	9,836	39.6	4,252	73.9	1,504	26.1	34,017	62.8	20,177	37.3
2003	8,817	66.6	4,430	33.4	6,317	59.0	4,391	41.0	15,008	60.5	9,802	39.5	4,408	74.4	1,545	25.9	34,550	63.1	20,168	36.9
2002	8,757	65.9	4,534	34.1	6,295	59.0	4,381	41.0	14,767	60.3	9,725	39.7	4,435	74.5	1,520	25.5	34,253	62.9	20,161	37.1
2001	9,007	65.6	4,720	34.4	6,412	58.8	4,489	41.2	15,374	60.1	10,219	39.9	4,538	73.5	1,633	26.5	35,331	62.7	21,062	37.3
2000	9,016	66.0	4,653	34.0	6,401	59.4	4,369	40.6	15,367	60.1	10,220	39.9	4,580	73.7	1,634	26.3	35,363	62.9	20,876	37.1
1999	8,821	66.2	4,514	33.8	6,331	59.7	4,273	40.3	15,253	60.4	10,000	39.6	4,376	74.0	1,536	26.0	34,781	63.1	20,323	36.9

Figure 5.5: Percentage of male female journalists

.

5.4 Conclusion

We built a model for gender prediction, by using the New York Times Annotated Corpus, assigning Male and Female labels to the articles. The corpus is split into 90 percent training and 10 percent testing. We perform 10 way cross validation on a sample of the dataset to select the model which best fits this corpus. Logistic Regression seems to performing the best for this specific corpus. We get accuracy of 85.81 percent on a sample of dataset and low recall scores. Accuracy score of 82.78 percent is obtained for both male and female labels on testing the model on the whole dataset the performance improves as we increase the threshold. The threshold is set to 65 percent to balance the recall and thereby increasing the performance of the learning model. The model predicts 79.61 percent articles to be authored by male and 20.27 percent documents to be authored by female.

Bibliography

[1] Shlomo Argamaon, Moshe Kopel, Anat Rachel *Automatically Categorizing Text on the basis of Gender*(Published in Literary and Linguistic Computing 17(4), 2002).

[2] Arjun Mukherjee, B Liu, *Improving gender classification of blog authors* (Conference on Empirical Methods in Natural Language Processing, 2010)

[3] Argamon, S., Koppel, M., J Fine, AR Shimoni. . , 2003. *Gender, genre, and writing style in formal written texts* (Text-Interdisciplinary Journal, 2003)

[4] ,Alexander Pak, Patrick Paroubek Twitter as a Corpus for Sentiment Analysis, *Twitter as a Corpus for Sentiment Analysis and Opinion Mining* (LREc. Vol. 10. No. 2010).

[5] Zhang, Cathy, and Pengyu Zhang. . *Predicting gender from blog posts."* University of Massachussetts Amherst, USA (2010).

[6] Argamon, S., Koppel, M., Pennebaker, J. W., Schler, J. .R.W. Boyd, *Automatically profiling the author of an anonymous text.*(Communications of the ACM, 52(2), 119-123 (2009).

[7] Yang, Yiming, and Jan O. Pedersen. " *A comparative study on feature selection in text categorization* (Icml. Vol. 97. 1997).

[8] McCallum, Andrew, and Kamal Nigam. ,*A comparison of event models for naive bayes text classification.* (AAAI-98 workshop on learning for text categorization. Vol. 752. No. 1. 1998)

[9] Bird, Steven, and Edward Loper*NLTK: the natural language toolkit.* (Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. Association for Computational Linguistics, 2004)

[10] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. . *The elements of statistical learning* (Vol. 1. New York: Springer series in statistics, 2001).

[11] Pedregosa, Fabian, et al. "Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.

[12] Pedregosa, Fabian, et al. *Scikit-learn: Machine learning in Python.* (Journal of machine learning research 12.Oct: 2825-2830) (2011).

[13] Liaw, Andy, and Matthew Wiener.*"Classification and regression by randomForest."* R news 2.3 : 18-22 (2002)

[14] Zhang, Tong, and Frank J. Oles. *"Text categorization based on regularized linear classification methods."*(Information retrieval 4.1 : 5-31. (2001)

[15] Shalev-Shwartz, Shai, and Shai Ben-David. *"Understanding machine learning: From theory to algorithms"*(Cambridge university press, 2014)

[16] Kivinen, Jyrki, and Manfred K. Warmuth.*"Exponentiated gradient versus gradient descent for linear predictors."* (Information and Computation 132.1 1-63 (1997)).

[17] Amy Joyce *"Is journalism really a male-dominated field? The numbers say yes"* (Washington Post 2014)

[18] New York Times Annotated Corpus *https://catalog.ldc.upenn.edu/ldc2008t19*

[19] G Louppe *"Understanding Random Forests: From Theory to Practice"*

[20] *Machinelearningmasterey.com*