GALGOTIAS UNIVERSITY

SCHOOL OF COMPUTING SCIENCE & ENGINEERING

PROGRESS REPORT ETE VIVA

Fall 2021-2022

B.Tech., / BCA / B.Sc., / M-Tech., / MCA / M.Sc.,

Project Details: **Semester:** 5th **Project ID:** BT3439

Project Title	NLP TEXT CLASSIFICATION USING MACHINE LEARNING WITH MARKOV CHAINING PROCESS
Progress of Project (in words)	Some . Module of code will not supporte. to Executed.
Research Paper Title	NLP TEXT CLASSIFICATION USING MACHINE ~~LE~~ LEARNING WITH MARKOV CHAINING PROCESS
Progress of Research Paper	Submitted to the Conferene

ETE VIVA DETAILS:

ETE VIVA DATE	ETE VIVA TIME	ROOM NO
24/12/21	1:50	C-434

Student Progress Details (Filled by Guide Only):

S. No	Name	Admission Number	No. Of time Came for Discussion	Performance of Student	Approval for Review 1
1	Rishabh Shakya	19SCSE1010262	2	□ Satisfactory ☑ Good □ Poor	☑ Approved □ Not Approved
2	Suryam Srivastava	19SCSE1180057	2	□ Satisfactory ☑ Good □ Poor	☑ Approved □ Not Approved
3				□ Satisfactory □ Good □ Poor	□ Approved □ Not Approved

24/12/2021

Guide Name & Signature with Date

24/12/21

Reviewer Name & Signature with Date

Scanned by TapScanner

A Thesis/Project/Dissertation Report

on

NLP TEXT CLASSIFICATION USING MACHINE LEARNING AND MARKOV CHAINING PROCESS

Submitted in partial fulfillment of the

requirement for the award of the degree of

Master of Computer Applications

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Name of Supervisor: Mr. DAMODHARAN D.
Assistant Professor
Department of Computer Science and Engineering

Submitted By

19SCSE1010262 - Rishabh Shakya

19SCSE1180057 - Suryam Srivastava

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING /
DEPARTMENT OF COMPUTERAPPLICATION
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER-2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"NLP TEXT CLASSIFICATION USING MACHINE LEARNING AND MARKOV CHAINING PROCESS"** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of **July-2021 to December-2021,** under the supervision of **Mr. Damodharan D. Assistant Professor,** Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

19SCSE1010262 - Rishabh Shakya

19SCSE1180057 - Suryam Srivastava

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Damodharan D.

Assistant Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **19SCSE1010262 – RISHABH SHAKYA, 19SCSE1180057 – SURYAM SRIVASTAVA** has been held on ________________ and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING.**

24/12/2021

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date:

Place:

Abstract

ABSTRACT Modern era is all about data. Since the dawn of internet from 1980 till 2003, 5 exabytes of data were generated. Now this much data is generated in every two days. Whether we surf the internet or listen to songs or scroll instagram, everywhere on internet we leave digital footprints. These digital footprints were gathered by various tech giants and a large amount of data is generated. Handling and managing of this data is the primary job of the data scientist. This useful information is implemented in the way that benefits and eases humans. The data scientists combine various machine learning algorithms to the data and extract and predict the useful information. Nowadays, high amount of data is generated in the form of text. Extracting the useful chunks of data from this huge class of text data is a difficult process. This is where our project comes into picture. The name of our project is "Text Classification". It combines the concept of natural processing language and Support Vector Machine (a ML algorithm) for the classification of various texts. It tags/classifies various portions of the text data into various categories such as whether the given text is talking affirmative about a topic or in some sort of negative sense, whether the given text is talking about electronics or aerodynamics or space etc. This helps us group the data into various categories and filter the data as per our requirements. This project would be of great help to data scientists and various MNC's which generates huge amount of data.

Table of Contents

List of Table

List of Figures

Acronyms

SVM	Support Vector Machine
ML	Machine Learning
DL	Deep Learning
CNN	Convolution Neural Networks

CHAPTER-1

Introduction

The name of our project is “TEXT CLASSIFICATION” .Our project is a machine learning/deep leaning enabled project whose main task is to cluster together various texts based on their meaning .For this purpose we will be using various ML/DL techniques. Text classification is a machine learning technique that assigns a set of predefined categories to open ended text. Text classifiers can be used to organize, structure, and categorize pretty much any kind of text- from documents, medical studies and files, and all over the web. For example, new articles can be organized by topics, support tickets can be organized by urgency, chat conversations can be organized by language ,brand mentions can be organized by sentiments and various other things .Text classification is one of the fundamental task in natural language processing with vast variety of broad applications such as sentiment analysis , topic labelling, spam detection ,intent detection and various other purposes.

CHAPTER-2

Literature Survey

Document fragmentation or document fragmentation is a problem in library science, information science and computer science. The task is to assign the document to one or more categories or categories. This can be done "practically" (or "intelligently") or algorithmically. The classification of literary genre has primarily become a provincial science library, while algorithmic classification is both information science and computer science. The problems are interdependent, however, so there is a cross-sectional study on the classification of texts.

Distributed texts can be text, images, music, etc. Each type of text has its own special classification problems. If otherwise stated, text splitting is implied.

Documents can be categorized according to their titles or by other factors (such as document type, author, year of publication etc.). Throughout this article only subject division is considered. There are two main philosophies for the division of titles: the content-based approach and the application-based approach.

CHAPTER-3

Project Design

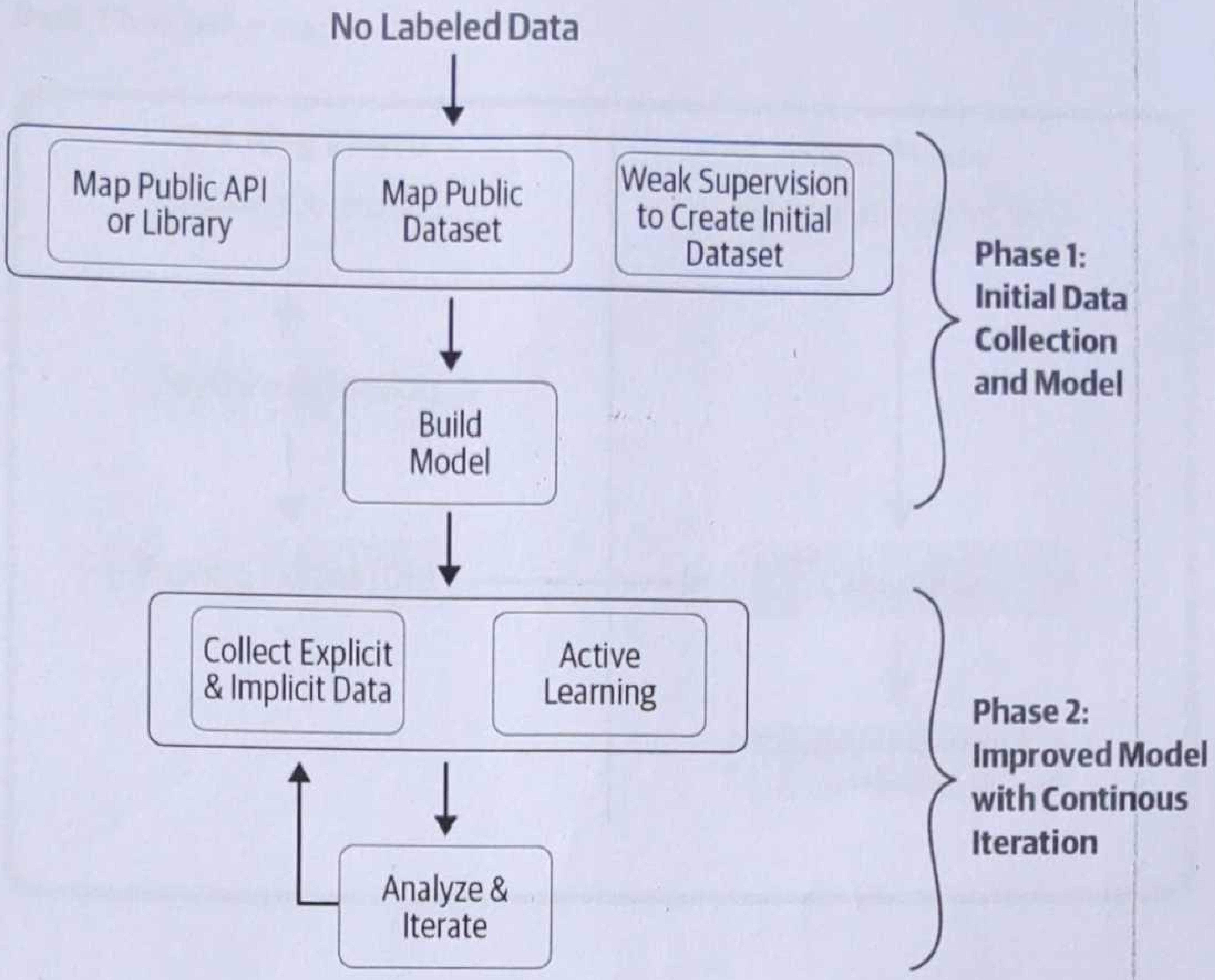
Use of Text Classifier:

It's estimated that around 80% of all information is unstructured, with text being one of the most common types of unstructured data. Because of the messy nature of text, analyzing, understanding, organizing, and sorting through text data is hard and time-consuming, so most companies fail to use it to its full potential.

This is where text classification with machine learning comes in. Using text classifiers, companies can automatically structure all manner of relevant text, from emails, legal documents, social media, chatbots, surveys, and more in a fast and cost-effective way. This allows companies to save time analyzing text data, automate business processes, and make data-driven business decisions.

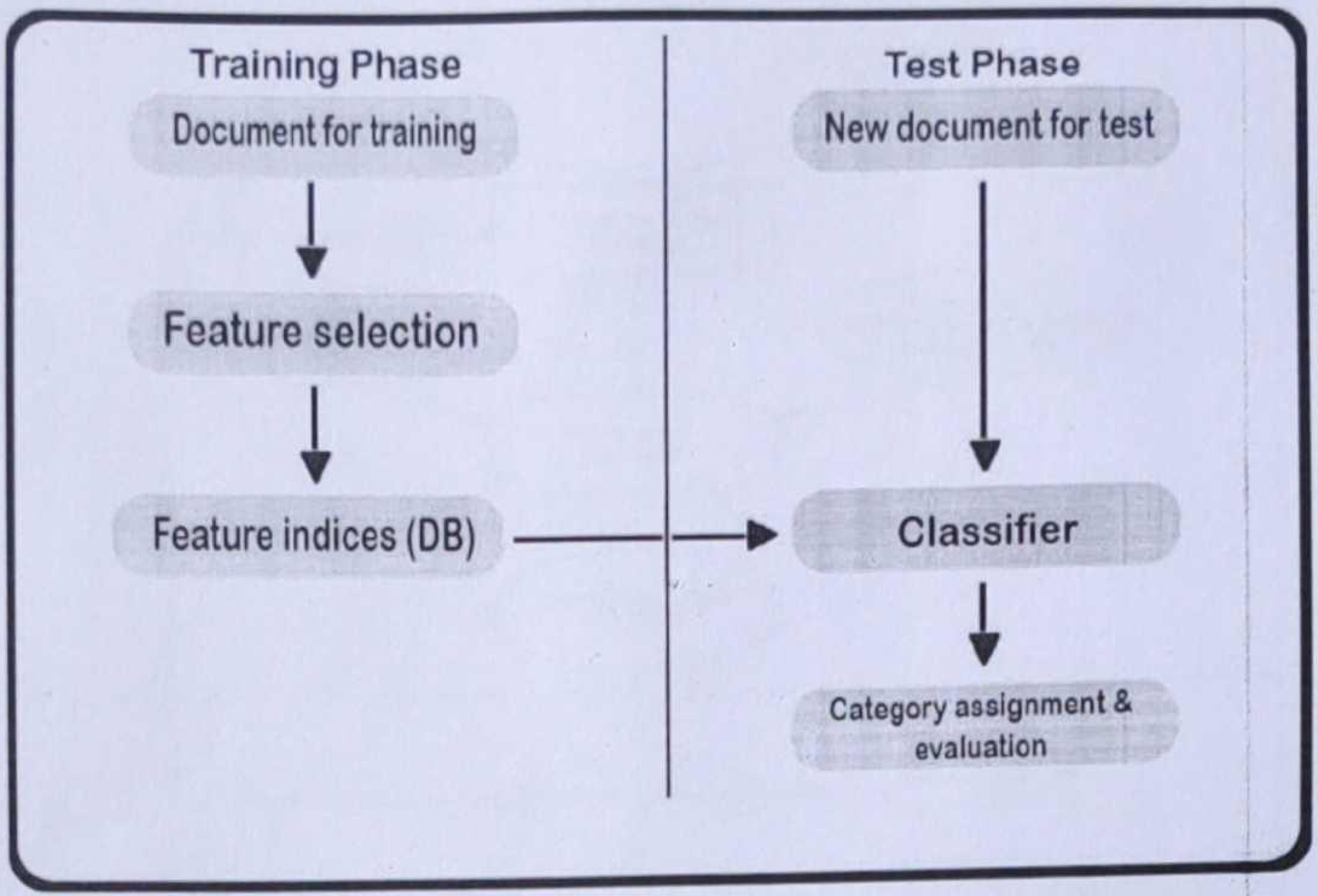
Building Of Text Classifier:

Automatic text classification applies machine learning, natural language processing (NLP), and other AI-guided techniques to automatically classify text in a faster, more cost-effective, and more accurate manner.

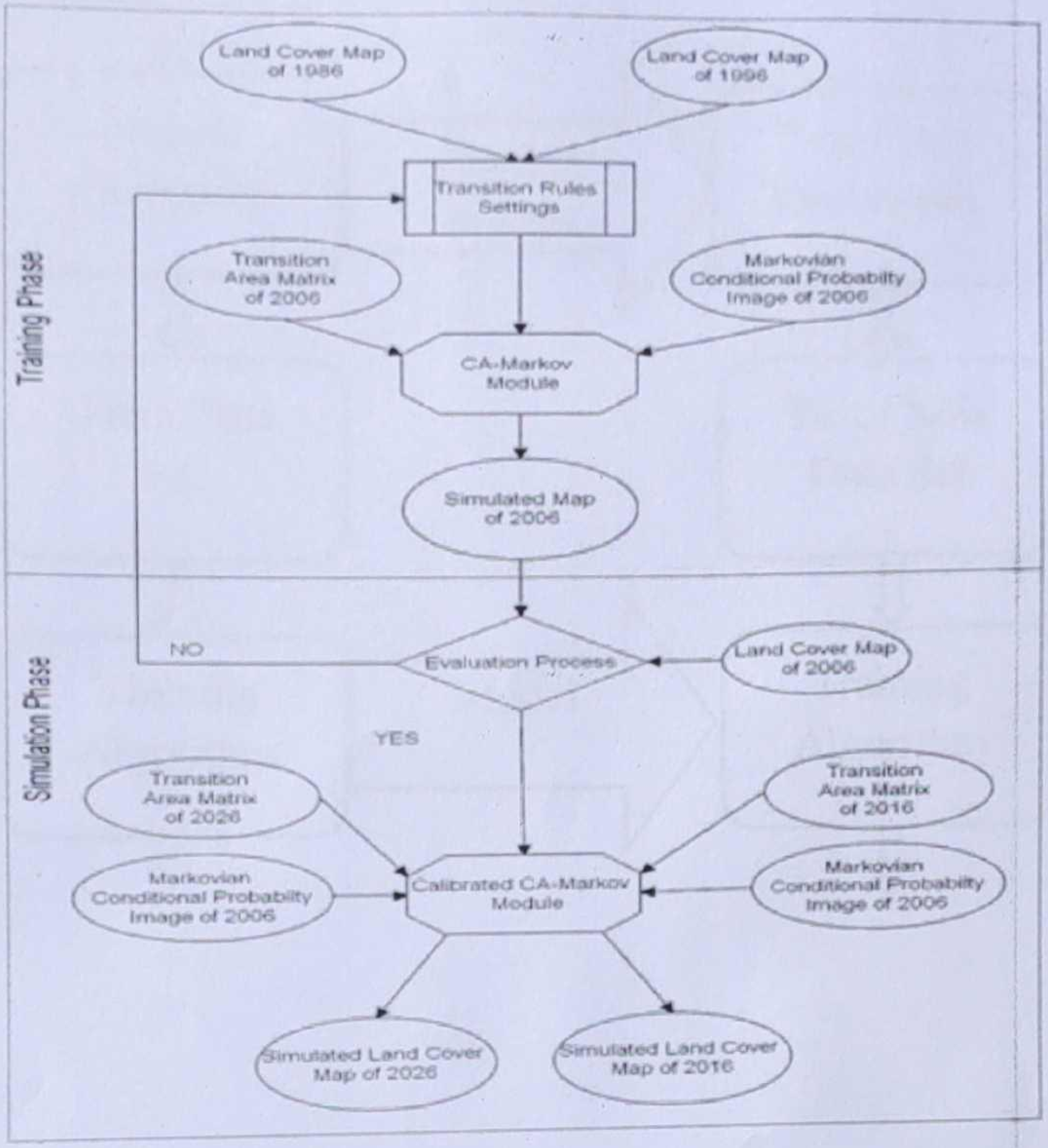

No Labeled Data
Map Public API or Library
Map Public Dataset
Weak Supervision to Create Initial Dataset
Build Model
Phase 1: Initial Data Collection and Model
Collect Explicit & Implicit Data
Active Learning
Analyze & Iterate
Phase 2: Improved Model with Continous Iteration

In this , we're going to focus on automatic text classification.

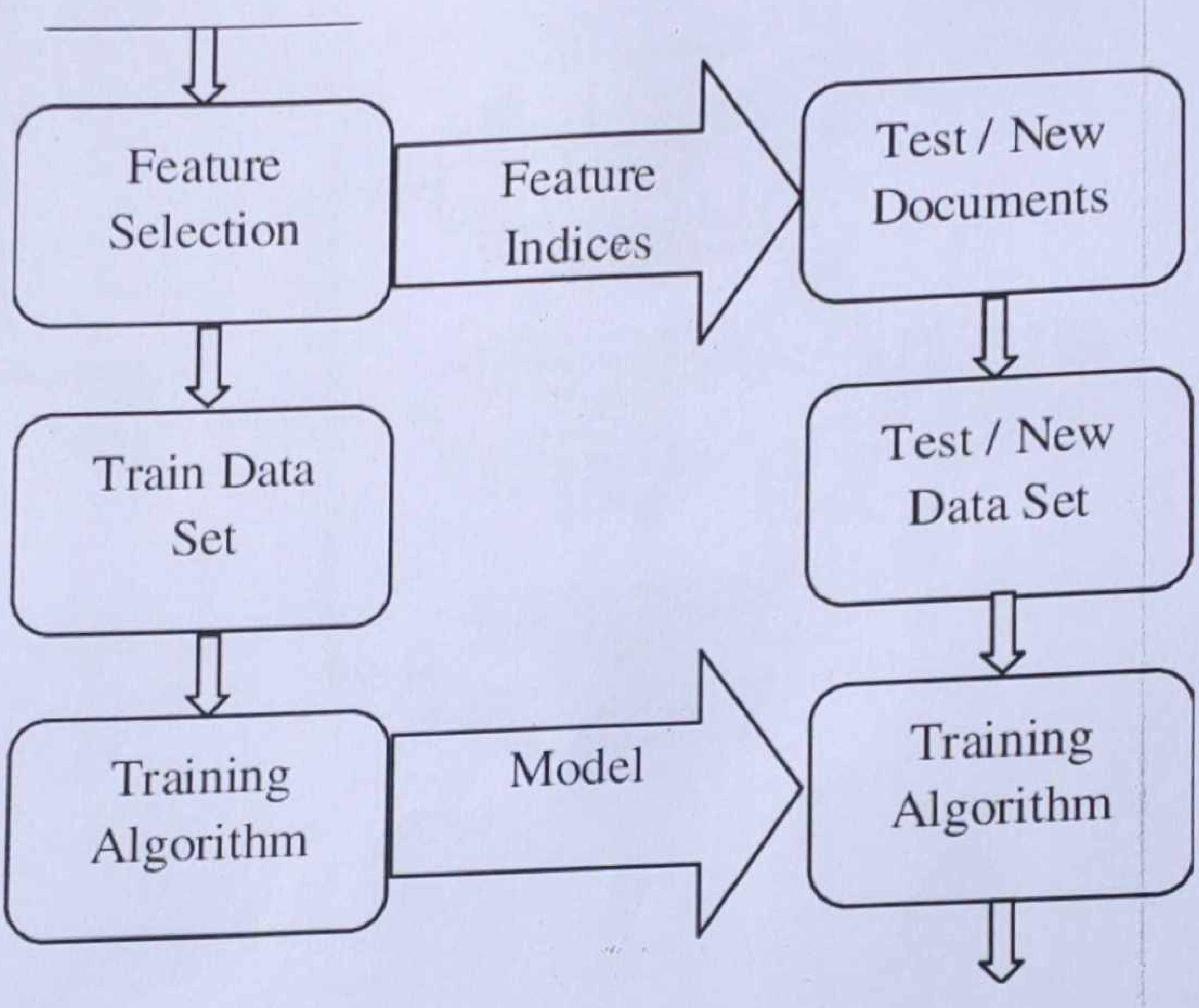
Data Flow Diagram:



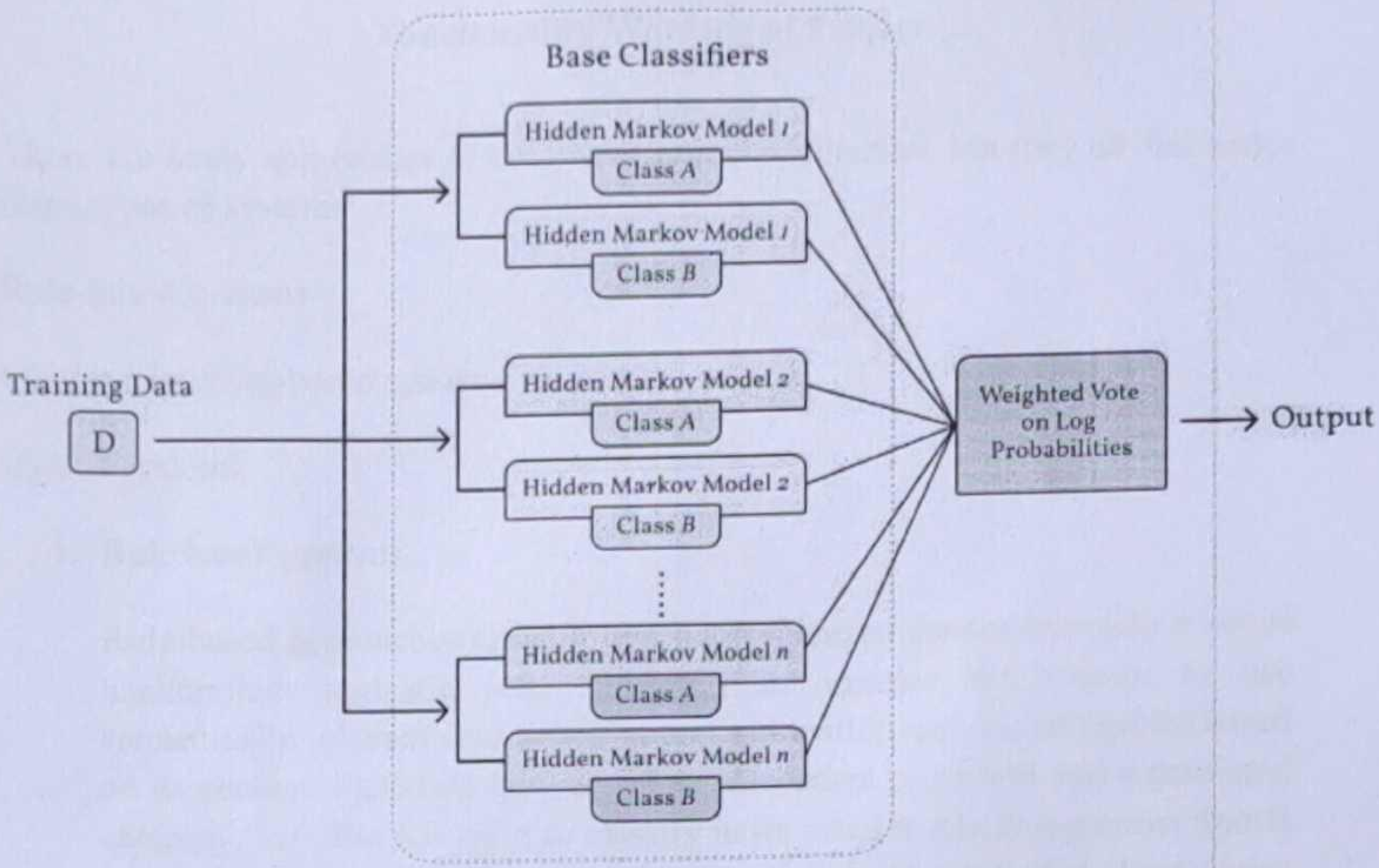
ER Diagram:


Training Phase
Land Cover Map of 1986
Land Cover Map of 1996
Transition Rules Settings
Transition Area Matrix of 2006
Markovian Conditional Probabilty Image of 2006
CA-Markov Module
Simulated Map of 2006
Simulation Phase
NO
Evaluation Process
Land Cover Map of 2006
YES
Transition Area Matrix of 2026
Transition Area Matrix of 2016
Markovian Conditional Probability Image of 2006
Calibrated CA-Markov Module
Markovian Conditional Probabilty Image of 2006
Simulated Land Cover Map of 2026
Simulated Land Cover Map of 2016

Flow Diagram:



Architecture Diagram:



CHAPTER-4

Functionality/Working of Project

There are many approaches to automatic text classification, but they all fall under three types of systems:

Rule-based systems

Machine learning-based systems

Hybrid systems

1. Rule based systems

   Rule-based approaches classify text into organized groups by using a set of handcrafted linguistic rules. These rules instruct the system to use semantically relevant elements of a text to identify relevant categories based on its content. Each rule consists of an antecedent or pattern and a predicted category. Say that you want to classify news articles into two groups: Sports and Politics. First, you'll need to define two lists of words that characterize each group (e.g., words related to sports such as football, basketball, LeBron James, etc., and words related to politics, such as Donald Trump, Hillary Clinton, Putin, etc.).Next, when you want to classify a new incoming text, you'll need to count the number of sport-related words that appear in the text and do the same for politics-related words. If the number of sports-related word appearances is greater than the politics-related word count, then the text is classified as Sports and vice versa.

   For example, this rule-based system will classify the headline "When is LeBron James' first game with the Lakers?" as Sports because it counted one sports-related term (LeBron James) and it didn't count any politics-related terms.

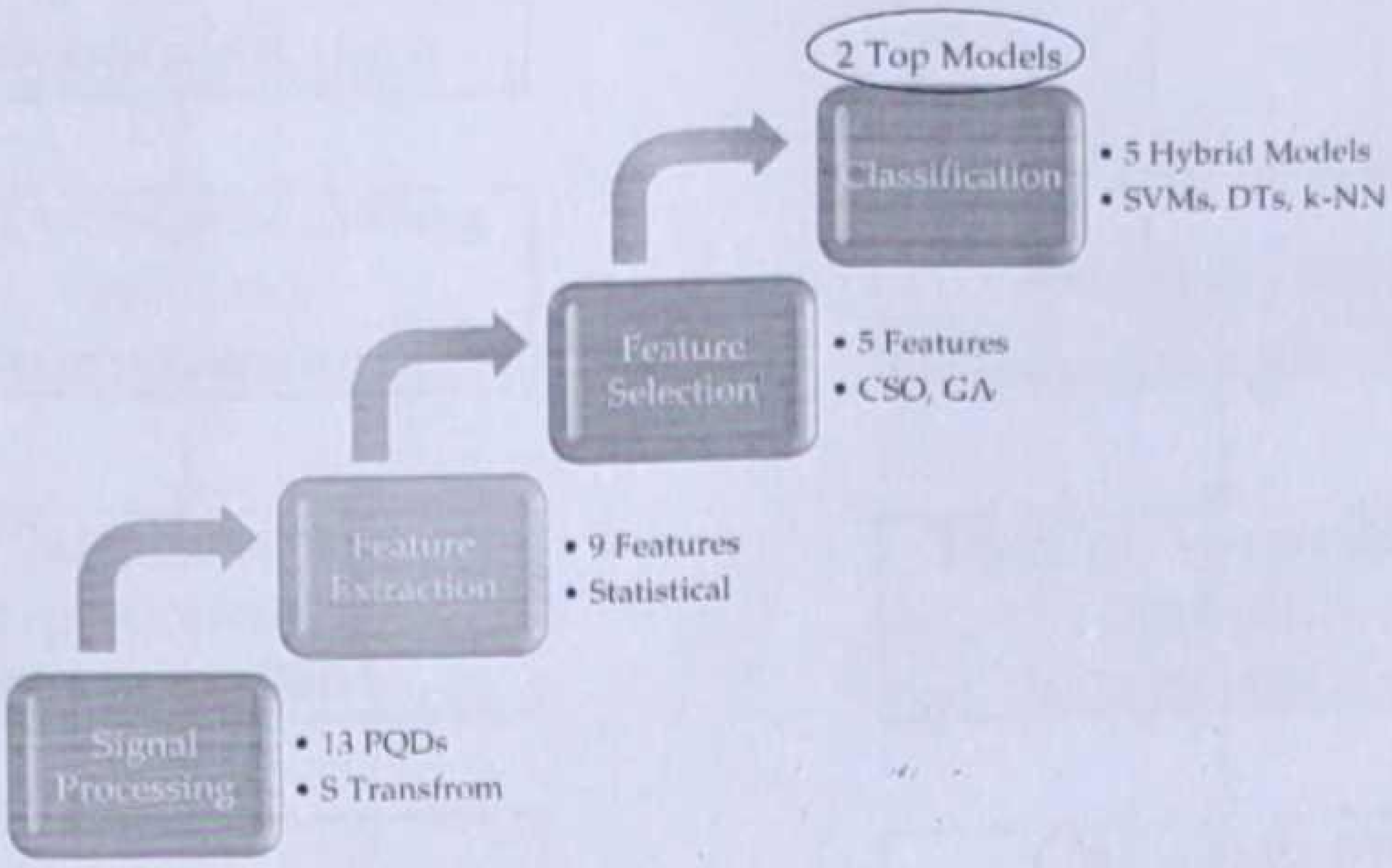
   Rule-based systems are human comprehensible and can be improved over time. But this approach has some disadvantages. For starters, these systems require deep knowledge of the domain. They are also time-consuming, since generating rules for a complex system can be quite challenging and usually requires a lot of analysis and testing. Rule-based systems are also difficult to maintain and don't scale well given that adding new rules can affect the results of the pre-existing rules.

2. Machine learning based systems

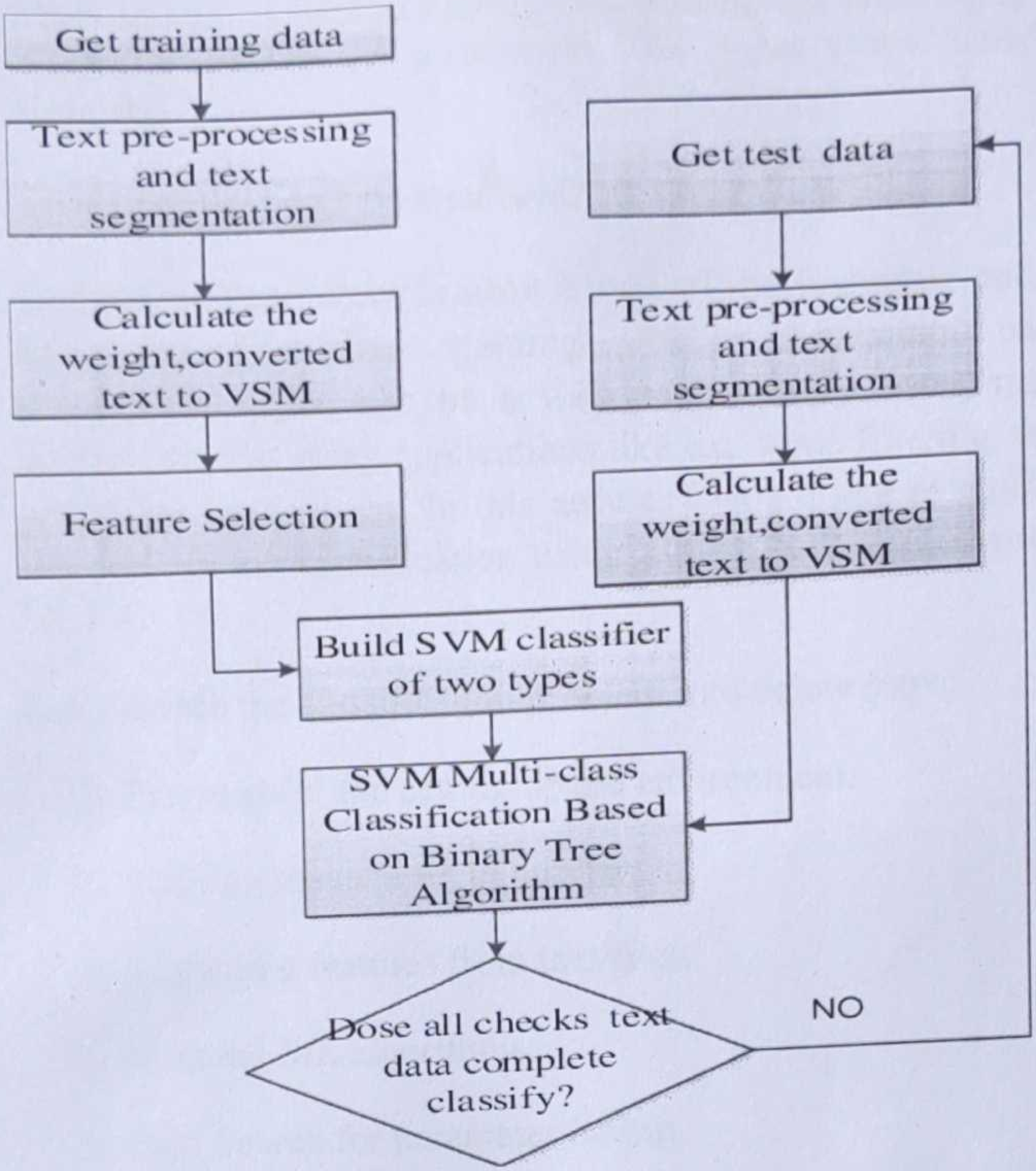
   Instead of relying on manually crafted rules, machine learning text classification learns to make classifications based on past observations. By using pre-labeled examples as training data, machine learning algorithms can learn the different associations between pieces of text, and that a particular output (i.e., tags) is expected for a particular input (i.e., text). A "tag" is the pre-determined classification or category that any given text count. For example, if we have defined our dictionary to have the following words {This, is, the, not, awesome, bad, basketball}, and we wanted to vectorize the text "This is awesome," we would have the following vector representation of that text: (1, 1, 0, 0, 1, 0, 0).Then, the machine learning algorithm is fed with training data that consists of pairs of feature sets (vectors for each text example) and tags (e.g. sports, politics) to produce a classification model.

3. Hybrid Systems

   Hybrid systems combine a machine learning-trained base classifier with a rule-based system, used to further improve the results. These hybrid systems can be easily fine-tuned by adding specific rules for those conflicting tags that haven't been correctly modeled by the base classifier.



Building process of the text classifier:


Get training data
Text pre-processing and text segmentation
Calculate the weight,converted text to VSM
Feature Selection
Get test data
Text pre-processing and text segmentation
Calculate the weight,converted text to VSM
Build SVM classifier of two types
SVM Multi-class Classification Based on Binary Tree Algorithm
Dose all checks text data complete classify?
NO

We will be primarily using Support Vector Machine algorithm for building of text classifier.

There are various machine learning, deep learning algorithms at play which we will discuss in detail in our presentation. This is just a brief introduction of our research/

Document/Text classification overview:

Document/Text classification is one of the important and typical task in *supervised* machine learning (ML). Assigning categories to documents, which can be a web page, library book, media articles, gallery etc. has many applications like e.g. spam filtering, email routing, sentiment analysis etc. In this article, I would like to demonstrate how we can do text classification using python, scikit-learn and little bit of NLTK.

Let's divide the classification problem into below steps:

1. Prerequisite and setting up the environment.
2. Loading the data set in jupyter.
3. Extracting features from text files.
4. Running ML algorithms.
5. Grid Search for parameter tuning.
6. Useful tips and a touch of NLTK.

Step 1: Prerequisite and setting up the environment

The prerequisites to follow this example are python version **2.7.3** and jupyter notebook. Also, little bit of python and ML basics including text classification is required. We will be using scikit-learn (python) librarie our example.

Step 2: **Loading the data set in jupyter.**

The data set will be using for this example is the famous “20 Newsgoup” set.

Step 3: Extracting features from text files.

Text files are actually series of words (ordered). In order to run machine learning algorithms we need to convert the text files into numerical feat vectors. We will be using **bag of words** model for our example. Briefly, segment each text file into words (for English splitting by space), and co # of times each word occurs in each document and finally assign each w an integer id. Each unique word in our dictionary will correspond to a feature (descriptive feature).

Step 4. Running ML algorithms.

There are various algorithms which can be used for text classification.

Performance of NB Classifier: Now we will test the performance of NB classifier on **test set.**

```
import numpy as np

twenty_test = fetch_20newsgroups(subset='test', shuffle=True)

predicted = text_clf.predict(twenty_test.data)

np.mean(predicted == twenty_test.target)
```

Naive Bayes

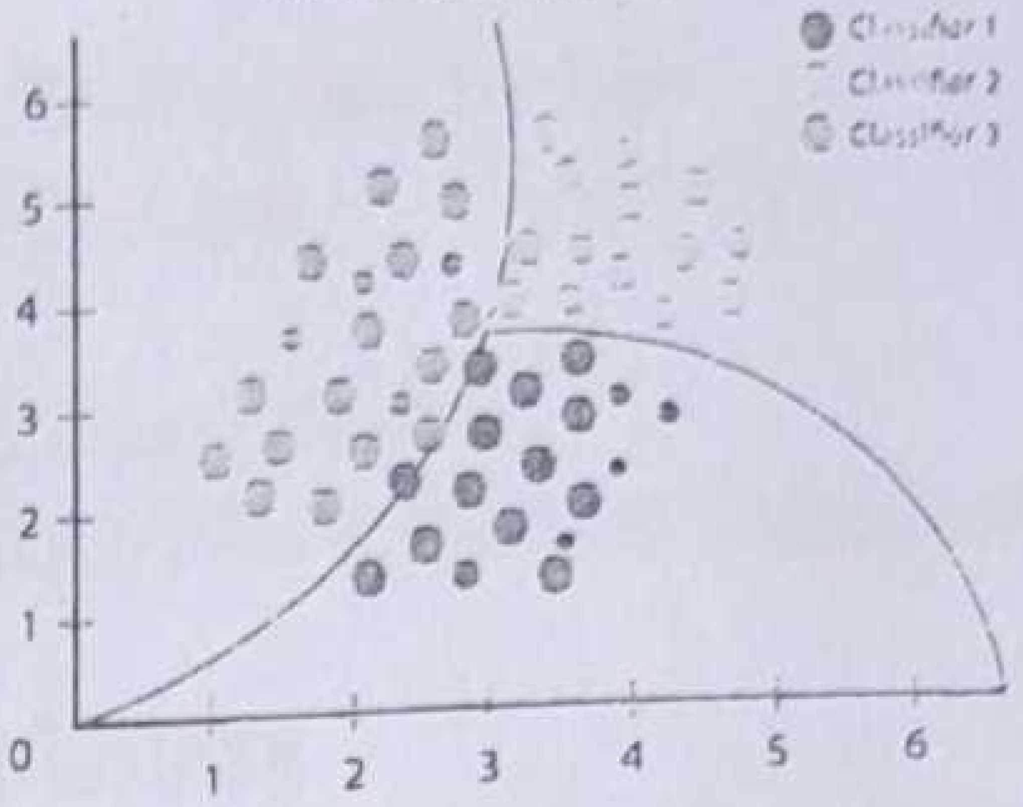
In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

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

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$

Naive bayes classifier



CHAPTER-5 Reference

Books:

1. Natural Language Processing with Python by Edward Loper, Ewan Klein, Steven Bird
2. NLP at Work by Sue Knight
3. Pattern Recognition and Machine Learning by Christopher Bishop
4. Neural Networks and Deep Learning by Charu C. Aggarwal Websites:

1. https://www.tensorflow.org/tutorials/keras/text_classification
2. https://towardsdatascience.com/deep-learning-techniques-for-textclassification-78d9dc40bf7c
3. https://realpython.com/python-keras-text-classification/

Altinçay, H., & Erenel, Z. (2010). Analytical evaluation of term weighting schemes for text categorization. Pattern Recognition Letters, 31, 1310–1323.

Altinçay, H., & Erenel, Z. (2012). Using the absolute difference of term occurrence probabilities in binary text categorization. ApplIntell, 36, 148–160.

Amine, B.M., & Mimoun, M. (2007).WordNet based cross-language text categorization. IEEE/ACS International Conference on Computer Systems and Applications, AICCSA.

An, J., & Chen, Y.P.P. (2005).Keyword Extraction for Text Categorization. Proceedings of the IEEE International Conference on Active Media Technology AMT.

Antonie, M.L., & Zai'ane, O.R. (2002). Text document categorization by term association. Proceedings of the IEEE International Conference on Data Mining, ICDM.

Azam, N., & Yao, J.T. (2012). Comparison of term frequency and document frequency based feature selection metrics in text categorization. Expert Systems with Applications, 39, 4760–4768.

Bakus, J., & Kamel, M.S. (2006). Higher order feature selection for text classification. Knowledge Information System, 9(4), 468-491.

Basu, A., Watters, C., & Shepherd, M. (2002). Support vector machines for text categorization. Proceedings of the 36th Hawaii IEEE International Conference on System, HICSS'03.

Cabrera, R.G., Gomez, M.M.Y., Rosso, P., & Pineda, L.V. (2009). Using the Web as corpus for selftraining text categorization. Information Retrieval, 12, 400–415.

Canfora, G., & Cerulo, L. (2005). How software repositories can help in resolving a new change request. Workshop on Empirical Studies in Reverse Engineering.

Catal, C. (2011). Software fault prediction: A literature review and current trends. Expert Systems with Applications, 38, 4626-4636.

Chang, Y.C., Chen, S.M., & Liau, C.J. (2008). Multi-label text categorization based on a new linear classifier learning method and a category-sensitive refinement method. Expert Systems with Applications, 34, 1948–1953.

Chen, E., Lin, Y., Xiong, H., Luo, Q., & Ma, H. (2011). Exploiting probabilistic topic models to improve text categorization under class imbalance. Information Processing and Management, 47, 202-214.

Chen, J., Zhou, X., & Wu, Z. (2004). A multi-label Chinese text categorization system based on boosting algorithm. Proceedings of the Fourth IEEE International Conference on Computer and Information Technology.

Chen, L., Guo, G., & Wang, K. (2011). Class-dependent projection based method for text categorization. Pattern Recognition Letters, 32, 1493-1501.

Chen, W., Yan, J., Zhang, B., Chen, Z., & Yang, Q. (2007). Document transformation for multilabel feature selection in text categorization. Seventh IEEE International Conference on Data Mining.

Chen, X.Y., Chen, Y., Wang, L., & Hu, Y.F. (2004). Text categorization based on frequent patterns with term frequency. Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August.

Thank You