VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**PHAN ANH TUAN – 521H0516**

**PHAM TRAN TIEN PHAT – 521H0285**

**NGUYEN VAN TRUONG – 521H0324**

**MID-TERM REPORT**

**INTRODUCTION TO  
NATURAL LANGUAGE PROCESSING**

**HO CHI MINH CITY, 2025**

VIETNAM GENERAL CONFEDERATION OF LABOUR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**PHAN ANH TUAN – 521H0516**

**PHAM TRAN TIEN PHAT – 521H0285**

**NGUYEN VAN TRUONG – 521H0324**

**MID-TERM REPORT**

**INTRODUCTION TO  
NATURAL LANGUAGE PROCESSING**

Instructor

**Assoc. Prof. Dr. Le Anh Cuong**

**HO CHI MINH CITY, 2025**

**ACKNOWLEDGEMENT**

We would like to thank Ton Duc Thang University, Faculty of Information Technology, for including this project in the training chart of Computer Science and for creating the best learning conditions for us as well as all students of the faculty. Our report is a product of what we have learned in the semester.

We would like to give **Mr. Le Anh Cuong**, who enthusiastically taught and worked tirelessly, to give me enough tools and skills to complete this report. He played an important role in improving my mathematical logic and knowledge. The second thanks I would like to give to the teachers of the Department of Information Technology of Ton Duc Thang University for giving me the opportunity to do this report, because it is not only an information technology project but also a very important experience for me in the next following years.

*Ho Chi Minh City, 19th March 2025*

*Author*

|  |
| --- |
|  |
|  |  |

*(Sign and write full name)*

*Phan Anh Tuan*

*Pham Tran Tien Phat*

*Nguyen Van Truong*

**DECLARATION OF AUTHORSHIP**

I hereby declare that this thesis was carried out by myself under the guidance and supervision of **Mr. Le Anh Cuong**. and that the work and the results contained in it are original and have not been submitted anywhere for any previous purposes. The data and figures presented in this thesis are for analysis, comments, and evaluations from various resources by my own work and have been duly acknowledged in the reference part.

In addition, other comments, reviews and data used by other authors, and organizations have been acknowledged, and explicitly cited.

**I will take full responsibility for any fraud detected in my thesis**. Ton Duc Thang University is unrelated to any copyright infringement caused on my work (if any).

*Ho Chi Minh City, 19th March 2025*

*Author*

|  |
| --- |
|  |
|  |  |

*(Sign and write full name)*

*Phan Anh Tuan*

*Pham Tran Tien Phat*

*Nguyen Van Truong*

**INSTRUCTOR VERIFICATION AND EVALUATION SECTION**

**Confirmation from the instructor**

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Ho Chi Minh City, 19th March 2025*

|  |
| --- |
|  |
|  |  |

*(Sign and write full name)*

**The teacher's evaluation part marks the test**

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Ho Chi Minh City, 19th March 2025*

|  |
| --- |
|  |
|  |  |

*(Sign and write full name)*

**TÓM TẮT**

Để giải quyết bài toán phân loại văn bản, nhóm chúng em có áp dụng các phương pháp biểu diễn văn bản truyền thống như Bag of Words, TF-IDF và N-gram, kết hợp với các thuật toán học máy như Naive Bayes, Logistic Regression, Decision Tree, Random Forest để phân tích dữ liệu.

Ngoài ra, nhóm chúng em cũng có triển khai phương pháp học sâu Doc2Vec, sử dụng cả mô hình đã được huấn luyện sẵn và mô hình tự huấn luyện từ đầu. Các phương pháp này được đánh giá và so sánh dựa trên ba yếu tố chính:

(1) Phương pháp biểu diễn văn bản.

(2) Tính đầy đủ và chính xác của dữ liệu.

(3) Hiệu quả của các phương pháp học máy.

Kết quả của nghiên cứu này giúp xác định phương pháp tối ưu để giải quyết bài toán phân loại cảm xúc trong văn bản, qua đó rút ra các khuyến nghị cho việc áp dụng trong các bài toán phân tích văn bản thực tế.

# ABSTRACT

To solve the text classification problem, our group has applied traditional text representation methods such as Bag of Words, TF-IDF and N-gram, combined with machine learning algorithms such as Naive Bayes, Logistic Regression, Decision Tree, Random Forest to analyze the data.

In addition, our group has also implemented the Doc2Vec deep learning method, using both pre-trained models and self-trained models from scratch. These methods are evaluated and compared based on three main factors:

(1) Text representation method.

(2) Completeness and accuracy of data.

(3) Efficiency of machine learning methods.

The results of this study help determine the optimal method to solve the problem of sentiment classification in text, thereby drawing recommendations for application in practical text analysis problems.

**INDEX**

[ABSTRACT… 2](#_Toc193665969)

[INDEX……… 3](#_Toc193665970)

[LIST OF FIGURES 3](#_Toc193665971)

[LIST OF ABBREVIATIONS 6](#_Toc193665972)

[CHAPTER 1 - INTRODUCTION 7](#_Toc193665973)

[1.1 Background 7](#_Toc193665974)

[1.2 Project Objectives 7](#_Toc193665975)

[CHAPTER 2 – DATA CONSTRUCTION 8](#_Toc193665976)

[2.1 Data Collection 8](#_Toc193665977)

[2.2 Exploratory Data Analysis 8](#_Toc193665978)

[2.3 Preprocessing data 9](#_Toc193665979)

[CHAPTER 3 – TRADITIONAL MACHINE LEARNING APPROACHES 12](#_Toc193665980)

[3.1 Text Representation Methods 12](#_Toc193665981)

[3.1.1 Bag of Words 12](#_Toc193665982)

[3.1.2 Term Frequency – Inverse Document Frequency 12](#_Toc193665983)

[3.1.3 N-grams 13](#_Toc193665984)

[3.2 Machine Learning Methods 14](#_Toc193665985)

[3.2.1 Decision Tree 14](#_Toc193665986)

[3.2.2 Navie Bayes 16](#_Toc193665987)

[3.2.3 Random Forest 18](#_Toc193665988)

[3.2.4 Support Vector Machine 20](#_Toc193665989)

[3.2.5 Logistic Regression 22](#_Toc193665990)

[CHAPTER 4 – DEEP LEARNING APPROACHES 25](#_Toc193665991)

[4.1 Introduction to Doc2Vec 25](#_Toc193665992)

[4.2 Training Doc2Vec from Scratch 26](#_Toc193665993)

[4.2.1 Introduction to the difference between Doc2Vec, LSTM, RNN and CNN 26](#_Toc193665994)

[4.2.2 Training Process 28](#_Toc193665995)

[4.2.3 Application and Evaluation 29](#_Toc193665996)

[4.3 Using Pre-trained Models 30](#_Toc193665997)

[4.3.1 Pre-trained Model Description 30](#_Toc193665998)

[4.3.2 Application and Evaluation 31](#_Toc193665999)

[CHAPTER 5 – COMPARATIVE ANALYSIS 34](#_Toc193666000)

[5.1 Based on Text Representation 34](#_Toc193666001)

[5.2 Based on Data Completeness and Accuracy 34](#_Toc193666002)

[5.3 Based on Machine Learning Methods 35](#_Toc193666003)

[5.4 Conclusions and Recommendations 36](#_Toc193666004)

[CHAPTER 6 – CONCLUSION 37](#_Toc193666005)

[REFERENCE 38](#_Toc193666006)

# 

# LIST OF FIGURES

[Figure 2. 1 Data Information 9](#_Toc193622417)

[Figure 2. 2 Labeling emotions 10](#_Toc193622418)

[Figure 2. 3 Load teencode dictionary 10](#_Toc193622419)

[Figure 2. 4 Convert emoticons to emoji 11](#_Toc193622420)

[Figure 3. 1 Bag of Words 13](#_Toc193622424)

[Figure 3. 2 Term Frequency – Inverse Document Frequency 14](#_Toc193622425)

[Figure 3. 3 N-grams 14](#_Toc193622426)

[Figure 3. 4 Decision Tree 15](#_Toc193622427)

[Figure 3. 5 Decision Tree + BoW 16](#_Toc193622428)

[Figure 3. 6 Decision Tree + TF-IDF 16](#_Toc193622429)

[Figure 3. 7 Decision Tree + N-gram(1,2) 17](#_Toc193622430)

[Figure 3. 8 Navie Bayes 17](#_Toc193622431)

[Figure 3. 9 Navie Bayes + BoW 18](#_Toc193622432)

[Figure 3. 10 Navie Bayes + TF-IDF 18](#_Toc193622433)

[Figure 3. 11 Navie Bayes + N-gram(1-2) 19](#_Toc193622434)

[Figure 3. 12 Random Forest 19](#_Toc193622435)

[Figure 3. 13 Random Forest + BoW 20](#_Toc193622436)

[Figure 3. 14 Random Forest + TF-IDF 20](#_Toc193622437)

[Figure 3. 15 Random Forest + N-gram (1,2) 21](#_Toc193622438)

[Figure 3. 16 Support Vector Machine 21](#_Toc193622439)

[Figure 3. 17 Support Vector Machine + BoW 22](#_Toc193622440)

[Figure 3. 18 Support Vector Machine + TF-IDF 22](#_Toc193622441)

[Figure 3. 19 Support Vector Machine + N-gram (1,2) 23](#_Toc193622442)

[Figure 3. 20 Logistic Regression 23](#_Toc193622443)

[Figure 3. 21 Logistic Regression + BoW 24](#_Toc193622444)

[Figure 3. 22 Logistic Regression + TF-IDF 24](#_Toc193622445)

[Figure 3. 23 Logistic Regression + N-gram (1,2) 25](#_Toc193622446)

[Figure 4. 1 Introduction to Doc2Vec 26](#_Toc193665939)

[Figure 4. 2 Distributed Memory (DM) 26](#_Toc193665940)

[Figure 4. 3 Distributed Bag of Words (DBOW) 27](#_Toc193665941)

[Figure 4. 4 LSTM - Long Short-Term Memory 28](#_Toc193665942)

[Figure 4. 5 RNN - Recurrent Neural Network 28](#_Toc193665943)

[Figure 4. 6 CNN - Convolutional Neural Network 29](#_Toc193665944)

[Figure 4. 7 Doc2Vec + Neural Network Classification 30](#_Toc193665945)

[Figure 4. 8 BERT 31](#_Toc193665946)

[Figure 4. 9 PhoBERT 32](#_Toc193665947)

[Figure 4. 10 BERT + Doc2Vec 33](#_Toc193665948)

[Figure 4. 11 PhoBERT + Doc2Vec 34](#_Toc193665949)

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| NLP | Natural Language Processing |
| TF-IDF | Term Frequency – Inverse Document Frequency |
| BoW | Bag of Words |
| SVM | Support Vector Machine |
| LSTM | Long Short-Term Memory |
| RNN | Recurrent Neural Network |
| CNN | Convolutional Neural Network |
| Doc2Vec | Document to Vector |
| BERT | Bidirectional Encoder Representations from Transformers |
| PhoBERT | A Vietnamese version of BERT |

# CHAPTER 1 - INTRODUCTION

## Background

Sentiment analysis from text is one of the extremely important problems in the field of Natural Language Processing (NLP). With the strong development of digital data and social networks, understanding and classifying emotions in text has become very necessary. Besides, the applications of sentiment analysis are also very diverse, combining from evaluating consumer feelings about products and services to analyzing social and political trends.

Current methods can be used to classify emotions according to many popular labels such as: sad, happy, angry, surprised, afraid, etc. From there, it can help businesses, organizations or researchers understand more deeply about feedback from the community.

## Project Objectives

The main goal is to apply traditional machine learning and deep learning methods to analyze and classify the type of sentiment in each text. Specifically, our team will:

* Apply traditional text representation methods such as Bag of Words, TF-IDF, N-gram and machine learning models such as Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine and Decision Tree.
* Use the Doc2Vec deep learning method with both pre-trained models and self-trained models.
* Compare the effectiveness of the above methods and draw conclusions about the optimal method for the problem of classifying emotions in text.

# CHAPTER 2 – DATA CONSTRUCTION

## 2.1 Data Collection

In the text sentiment analysis problem, dataset plays an essential role in the way for project works. Our team uses this dataset collect from **University of Information Technology** with the construction included nearly 7.000 rows (gained from a well-known social media platform, called **Facebook**) together with 2 specific columns, namely: *Emotion* and *Sentence*

**Ease of access:** social media and online platforms are places where users frequently share their emotions, providing a rich and diverse source of emotion data.

**Reality:** Comments on these platforms accurately reflect users' feelings, attitudes, and responses to events, products, or services.

**Emotion diversity:** Users can express a wide range of emotions such as happiness, sadness, anger, surprise, and many others, creating a rich dataset for the classification problem.

## 2.2 Exploratory Data Analysis

After collecting the data, the next step is to do a preliminary analysis to understand the dataset. Preliminary analysis helps to assess the quality of the data, discover the salient features, and help identify the elements that need improvement. Here are the initial data analysis results:

**Distribution of sentiment labels:** First, we need to check the distribution of sentiment labels in the dataset. We can see that the number of comments belonging to the emotions "Enjoyment" and "Sadness" may account for a large proportion, while other emotions such as "Fear" or "Anger" may be less.

**Sentence count:** Analyzing the number of sentences in the dataset helps to assess the length of the texts, helping to determine whether the data needs to be truncated or enriched.

**Special features:** Analyzing common words in each sentiment class helps to understand the semantics and keywords that characterize each sentiment

**Handling missing values:** Check the data for any missing information and determine how to handle missing values ​​(e.g., remove or replace).

## 2.3 Preprocessing data

**Step 1:** Install the necessary libraries

Install and import libraries needed for text and data processing such as **pandas**, **underthesea**, **emoji**, **demoji**, **langdetect**.

**Step 2:** Import library

• Import libraries needed for text processing, such as:

* **re** (regular expressions)
* **emoji** (emoji processing)
* **demoji** (emoji removal)
* **langdetect** (detect language)
* **underthesea** (Vietnamese word separation)

**Step 3:** Load and read data

A screenshot of a chat

AI-generated content may be incorrect.

Figure 2. Data Information

**Step 4:** Check for missing data

**Step 5:** Handling duplicate data

**Step 6:** Explore emotion labels

**Step 7:** Labeling emotions

**A screenshot of a phone

AI-generated content may be incorrect.**

Figure 2. Labeling emotions

**Step 8:** Load teencode dictionary

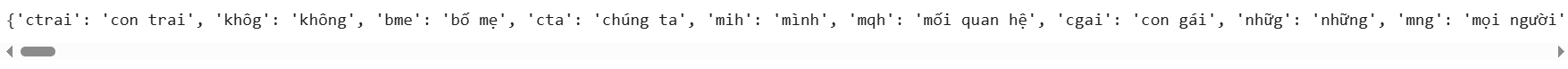
****

Figure 2. Load teencode dictionary

**Step 9:** Convert emoticons to emoji

**A screenshot of a video chat

AI-generated content may be incorrect.**

Figure 2. Convert emoticons to emoji

**Step 10:** Clean hashtag

**Step 11:** Remove short word

**Step 12:** Preprocess text

# CHAPTER 3 – TRADITIONAL MACHINE LEARNING APPROACHES

## 3.1 Text Representation Methods

To model text in sentiment analysis, we need to convert text into forms that computers can process. Common text representation methods are:

### 3.1.1 Bag of Words

This method represents a text as an unordered set of words. Each document is converted into a vector of word frequencies, where each value is the frequency of the word in the text.

A bag with words and a arrow

AI-generated content may be incorrect.This method is simple but may lack semantic information about the word in context.

Figure 3. Bag of Words

### 3.1.2 Term Frequency – Inverse Document Frequency

This method improves on BoW by weighting the importance of each word in a document relative to the entire dataset.

A computer screen shot of a keyboard

AI-generated content may be incorrect. Important words that appear frequently in a document but rarely appear in other documents will have a higher weight. This method helps reduce the influence of stop words such as "the", "is", "in".

Figure 3. Term Frequency – Inverse Document Frequency

### 3.1.3 N-grams

As an extension of BoW, N-grams considers sequences of consecutive words in a text instead of just considering individual words.

A row of words on a white background

AI-generated content may be incorrect.For example, with 2-grams, we will consider pairs of consecutive words (bigrams) in a text such as "happy day" or "sad movie". This method helps to preserve better semantics in the text.

Figure 3. N-grams

## 3.2 Machine Learning Methods

Once we have represented the text, we will apply machine learning methods to classify the sentiment in the text. Popular machine learning algorithms include:

### 3.2.1 Decision Tree

Decision Tree is a classification algorithm that builds a decision tree, where each node is a question about a feature (a word in a document) and branches based on the answer. Decision Trees are easy to understand and intuitive, making it easy for users to follow the decision-making process.

A diagram of a diagram

AI-generated content may be incorrect.However, Decision Trees can be overfitted if not pruned properly. When the tree is too complex, it can learn too many details of the training data, reducing its ability to generalize to new data. Pruning the tree helps to mitigate this problem and improve generalization.

Figure 3. Decision Tree

**-Evaluate Decision Tree with:**

A screenshot of a score

AI-generated content may be incorrect.+Bag of Words:

Figure 3. Decision Tree + BoW

A screenshot of a computer

AI-generated content may be incorrect.+TF-IDF:

Figure 3. Decision Tree + TF-IDF

A screenshot of a computer screen

AI-generated content may be incorrect.+N-gram(1,2)

Figure 3. Decision Tree + N-gram(1,2)

### 3.2.2 Navie Bayes

A diagram of a bar graph

AI-generated content may be incorrect.Naive Bayes is a classification algorithm based on Bayesian probability theory, which assumes that features (such as words in a text) are independent of each other. This means that each word is considered to have no influence on other words in the text, even though in reality words may be related to each other. Although this assumption is not always correct, Naive Bayes still performs well in many text classification problems, especially when the data is noisy.

Figure 3. Navie Bayes

The advantages of Naive Bayes are that it is simple, fast, and easy to implement, making it a popular choice in text classification applications. It is very effective in handling large data sets and can achieve good results even when some data is missing or imperfect. Although the independence assumption between features may not always hold, Naive Bayes often gives good results in practice.

**-Evaluate Navie Bayes with:**

A screenshot of a computer screen

AI-generated content may be incorrect.+Bag of Words:

Figure 3. Navie Bayes + BoW

A screenshot of a computer screen

AI-generated content may be incorrect.+TF-IDF

Figure 3. Navie Bayes + TF-IDF

A screenshot of a computer screen

AI-generated content may be incorrect.+N-gram(1-2)

Figure 3. Navie Bayes + N-gram(1-2)

### 3.2.3 Random Forest

Random Forest is a powerful machine learning method, belonging to the ensemble learning group, capable of classification and regression. This model uses a set of decision trees and combines their results to make the final prediction.

A diagram of a tree

AI-generated content may be incorrect.The main idea of ​​Random Forest is to build many independent decision trees and combine their results to improve accuracy and reduce overfitting. Using many decision trees helps the model avoid overfitting the training data, while improving the ability to generalize when predicting on new data.

Figure 3. Random Forest

Random Forest is very effective in handling complex data and can work well with many different types of data without too much fine-tuning.

**-Evaluate Random Forest with:**

A screenshot of a computer screen

AI-generated content may be incorrect.+Bag of Words:

Figure 3. Random Forest + BoW

A screenshot of a computer screen

AI-generated content may be incorrect.+TF-IDF

Figure 3. Random Forest + TF-IDF

A screenshot of a computer screen

AI-generated content may be incorrect.+N-gram(1-2)

Figure 3. Random Forest + N-gram (1,2)

### 3.2.4 Support Vector Machine

SVM is a powerful machine learning algorithm, commonly used in text classification, such as sentiment classification. This model searches for an optimal hyperplane to divide the text into different classes.

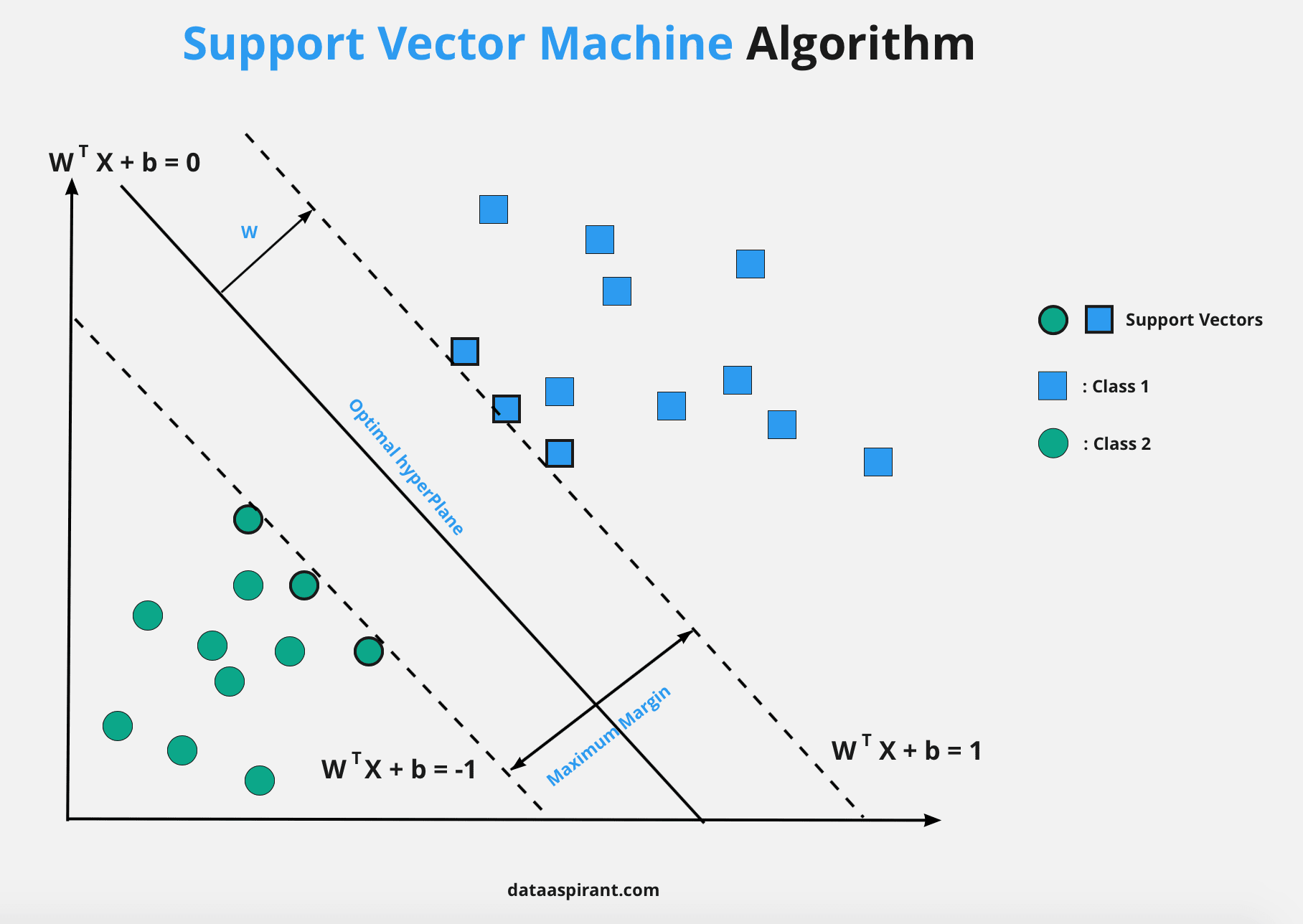
Although SVM is a linear classifier, it can handle nonlinear problems thanks to the kernel trick technique, which helps classify complex texts by moving data into high-dimensional space.

Figure 3. Support Vector Machine

SVM is often applied in NLP to classify text based on features such as TF-IDF, word frequency, or other semantic features, and gives accurate results, especially when the data has a complex structure.

**-Evaluate Random Forest with:**

A screenshot of a computer screen

AI-generated content may be incorrect.+Bag of Words:

Figure 3. Support Vector Machine + BoW

A screenshot of a computer screen

AI-generated content may be incorrect.+TF-IDF:

Figure 3. Support Vector Machine + TF-IDF

A screenshot of a computer

AI-generated content may be incorrect.+N-gram(1-2):

Figure 3. Support Vector Machine + N-gram (1,2)

### 3.2.5 Logistic Regression

Logistic Regression is a linear classification method used to predict the probability of an instance belonging to a particular class. It is widely used in classification problems where the relationship between the features and the target class is linear.

A graph of a logistic regression curve

AI-generated content may be incorrect.In NLP, Logistic Regression is particularly effective for tasks like sentiment analysis, where the goal is to classify text into categories such as positive, negative, or neutral.

Figure 3. Logistic Regression

By modeling the probability of each class, Logistic Regression provides a simple yet powerful tool for sentiment classification, especially when the data shows a linear relationship.

**-Evaluate Logistic Regression with:**

+Bag of Words:

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 3. Logistic Regression + BoW

+TF-IDF:

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 3. Logistic Regression + TF-IDF

+N-gram(1-2):

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3. Logistic Regression + N-gram (1,2)

# CHAPTER 4 – DEEP LEARNING APPROACHES

## 4.1 Introduction to Doc2Vec

A diagram of a machine

AI-generated content may be incorrect.Doc2Vec is a deep learning model developed based on the Word2Vec method, but instead of learning vectors for individual words, Doc2Vec learns vectors for the entire document. The goal of this method is to convert a long text into a fixed vector, which helps maintain the semantics of the entire text in a vector space.

Figure 4. Introduction to Doc2Vec

Doc2Vec uses a similar method to Word2Vec, including two main models:

* A diagram of a cat and a cat

  AI-generated content may be incorrect.**Distributed Memory (DM):** This method is like the Continuous Bag of Words (CBOW) model in Word2Vec, which helps learn vectors representing the text based on the context of surrounding words.

Figure 4. Distributed Memory (DM)

* A diagram of a cat and a cat

  AI-generated content may be incorrect.**Distributed Bag of Words (DBOW):** This method is similar to the Skip-Gram model in Word2Vec, but it teaches to predict words in a document from the vector of that text only, without the need for surrounding word context.

Figure 4. Distributed Bag of Words (DBOW)

By this way, each document can be represented by a fixed vector, making it easier for the model to handle text classification problems such as sentiment analysis.

## 4.2 Training Doc2Vec from Scratch

### 4.2.1 Introduction to the difference between Doc2Vec, LSTM, RNN and CNN

**Doc2Vec:** Is a deep learning model used to convert text into fixed vectors that represent the entire text. This model is very effective when you need to convert text into vectors for classification or sentiment analysis.

A diagram of a memory

AI-generated content may be incorrect.**LSTM (Long Short-Term Memory):** Is an improved form of RNN that helps overcome the vanishing gradient problem, very powerful in learning long-term r Delationships in sequence data.

Figure 4. LSTM - Long Short-Term Memory

A diagram of a network

AI-generated content may be incorrect.**RNN (Recurrent Neural Network):** Is a recurrent neural network model, used to process data sequences, but has limitations in processing long sequences due to the vanishing gradient problem.

Figure 4. RNN - Recurrent Neural Network

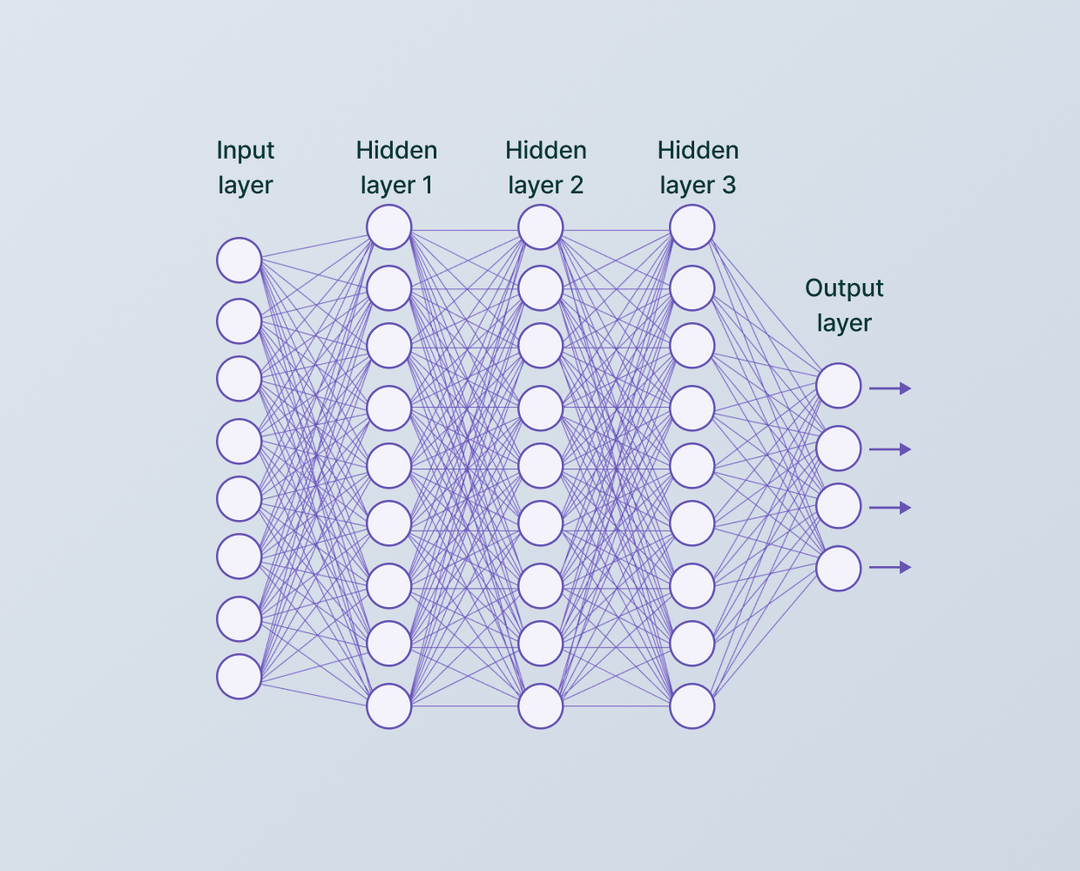
**CNN (Convolutional Neural Network):** Although CNN was mainly developed for image recognition, it can also be used for text classification, especially in identifying local features such as important phrases.

Figure 4. CNN - Convolutional Neural Network

### 4.2.2 Training Process

In this section, we will go through the steps of training Doc2Vec from scratch, starting from data preparation, model structure to optimization and training the model.

**1. Doc2Vec Model Structure:**

* **Vector Size:** Determines the number of dimensions of the vector representing each document. The common dimensions are 100-300 dimensions.
* **Hidden Layers:** Select the number of hidden layers in the model, this number of layers can affect the complexity and learning ability of the model.
* **Optimization Algorithm:** Use algorithms such as Stochastic Gradient Descent (SGD) or Adam to update the model weights during training.

**2. Training Process:**

* **Weight Initialization:** The model weights are initialized randomly or by a method such as Xavier or He initialization.
* **Weight Update:** After each training round, the model weights will be updated to minimize the error (loss). This process continues over many rounds (epochs) until the optimal result is achieved.
* **Learning algorithm:** In each epoch, the model learns from the documents and updates the weights so that the error is reduced. Make sure to choose a reasonable learning rate to avoid the model learning too fast or too slow.

### 4.2.3 Application and Evaluation

**1. Doc2Vec Model Applications:**

* **Text Sentiment Classification:** After training the model, you can use the learned text vectors to classify comments or posts into sentiment classes such as “Happy”, “Sad”, “Angry”, etc.
* **Document Search and Classification:** These vectors can be used to search for similar documents or classify them into appropriate groups.

**A screenshot of a computer screen

AI-generated content may be incorrect.2.** **Evaluation Doc2Vec:**

Figure 4. Doc2Vec + Neural Network Classification

## 4.3 Using Pre-trained Models

### 4.3.1 Pre-trained Model Description

Pretrained models are deep learning models that are pre-trained on large datasets and can be reused for other problems without having to train them from scratch. Models like BERT and Pho-BERT have been trained on large datasets and are capable of understanding the semantics of words and sentences.

* **BERT (Bidirectional Encoder Representations from Transformers):** BERT is a powerful deep learning model that is trained to understand the semantics of text in a bidirectional context (left-to-right and right-to-left). This helps the model better understand the relationship between words in a sentence and the context in which they appear.

A diagram of a diagram

AI-generated content may be incorrect.

Figure 4. BERT

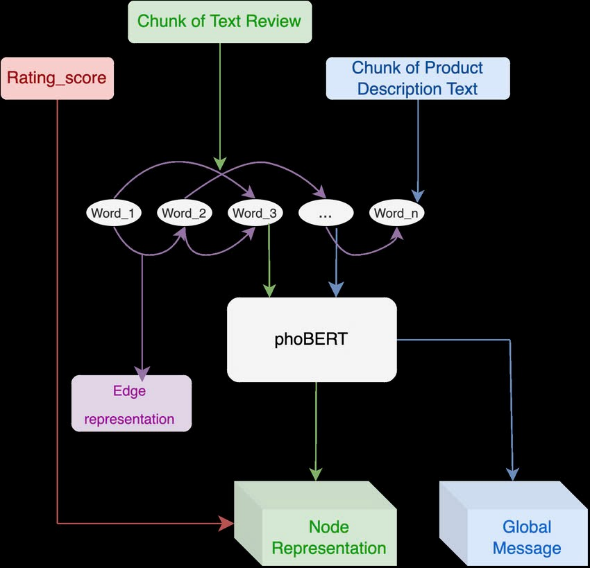
* **PhoBERT:** PhoBERT is an improved version of BERT, trained specifically on Vietnamese data. This model is designed to understand the semantics and structure of Vietnamese, helping to effectively solve NLP problems in Vietnamese, such as text sentiment analysis, text classification, and entity recognition.

Figure 4. PhoBERT

### 4.3.2 Application and Evaluation

**1. Applications of BERT and PhoBERT**

*BERT* and *PhoBERT* are powerful deep learning models that are pre-trained on large datasets and can be used to solve NLP problems such as text sentiment classification, entity recognition (NER), question answering (QA), etc. Here are some applications of these models:

* **Text sentiment classification:** BERT and PhoBERT can be used to classify texts into sentiment classes such as "Happy", "Sad", "Angry", etc. These models can understand the context in the text and accurately classify the sentiment regardless of the length of the text or the complexity of the context.

For example: Analyzing consumer sentiment on social media or in product and service reviews.

* **Named Entity Recognition (NER):** BERT and PhoBERT are capable of recognizing entities in text such as names of people, places, organizations, etc., which helps in information extraction problems.

For example: Extracting information from articles, legal documents, or press articles.

* **Question Answering:** Models like BERT and PhoBERT can be applied in question answering systems, where users ask questions, and the model answers based on the given textual context.

For example: Chatbots, automated customer support systems.

A screenshot of a graph

AI-generated content may be incorrect.**2. Evaluation of BERT and PhoBert**

Figure 4. BERT + Doc2Vec

A black and white table with numbers

AI-generated content may be incorrect.

Figure 4. PhoBERT + Doc2Vec

# CHAPTER 5 – COMPARATIVE ANALYSIS

## 5.1 Based on Text Representation

In this section, we compare the effectiveness of various text representation methods used in sentiment analysis. We evaluate traditional methods such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and N-grams, as well as deep learning methods like Doc2Vec.

* **Bag of Words** is a simple and intuitive method but lacks semantic context, which can lead to limited accuracy in certain cases. It performs well in tasks where exact word occurrences are important but struggles with capturing the overall meaning of the text.
* **TF-IDF** improves upon BoW by weighing words based on their importance across documents. It helps reduce the influence of common but uninformative words (e.g., "the," "is"), making it more effective in capturing meaningful content in text.
* **N-grams** provide better context by considering sequences of words rather than individual words. This method improves accuracy in tasks that require understanding the relationship between consecutive words.
* **Doc2Vec**, a deep learning method, transforms entire documents into fixed-length vectors, preserving semantic meaning. It captures the global structure of the text and is particularly useful in more complex sentiment analysis tasks.

## 5.2 Based on Data Completeness and Accuracy

The quality and completeness of the data are critical factors in the effectiveness of sentiment analysis models. In this report, we used data from social media comments and user responses, which reflect diverse emotional expressions.

* **Data Completeness**: The dataset includes a variety of emotions, including happiness, sadness, anger, surprise, and fear. However, some emotions like "fear" or "anger" were less represented. More balanced data across emotions is recommended for improved accuracy.
* **Accuracy**: The accuracy of the models was influenced by the quality of data preprocessing and feature extraction. Models using TF-IDF and N-grams performed better due to their ability to capture more relevant features from the text. Deep learning models like Doc2Vec showed even higher accuracy by leveraging semantic meaning and context.

## 5.3 Based on Machine Learning Methods

Various machine learning methods were used to analyze sentiment in text, including Decision Trees, Naive Bayes, Random Forest, Support Vector Machines (SVM), and Logistic Regression.

* **Decision Trees**: Decision Trees are simple and interpretable, but they tend to overfit if not pruned properly. They perform well with smaller datasets but may struggle with larger, more complex datasets.
* **Naive Bayes**: Despite its strong assumptions of feature independence, Naive Bayes is efficient and works well for text classification tasks, particularly when the data is noisy or imperfect.
* **Random Forest**: This ensemble learning method outperformed Decision Trees due to its ability to reduce overfitting. It is highly effective in handling complex datasets and provides more accurate predictions.
* **SVM**: SVM demonstrated strong performance, especially with high-dimensional data like TF-IDF or N-grams. It is effective in separating complex text categories but can be computationally expensive for large datasets.
* **Logistic Regression**: A linear classifier that performs well when there is a clear separation between sentiment classes. It is simple and efficient but may not handle complex, non-linear relationships in the data as well as other methods.

## 5.4 Conclusions and Recommendations

In conclusion, **Random Forest** and **SVM** demonstrated the best performance in this study, followed by **Logistic Regression** and **Naive Bayes**. For text representation, **Doc2Vec** outperformed traditional methods like BoW, TF-IDF, and N-grams by capturing the semantic meaning of the text. The combination of **TF-IDF** with **Random Forest** or **SVM** showed strong results for sentiment classification.

Recommendations for future improvements include:

* Expanding the dataset to include a more balanced representation of all emotional categories.
* Using more advanced models like **BERT** or **PhoBERT**, especially for languages like Vietnamese, which may improve performance.
* Further optimizing data preprocessing techniques to handle noisy and incomplete data more effectively.

# CHAPTER 6 – CONCLUSION

In this project, we have successfully applied various text representation methods such as Bag of Words, TF-IDF, and N-grams, along with machine learning models like Naive Bayes, Logistic Regression, Random Forest, Support Vector Machine (SVM), and Decision Tree, to classify emotions in text data. Additionally, we explored the application of deep learning methods like Doc2Vec, both with pre-trained models and self-trained models.

The main objective of this project was to determine the most effective approach for sentiment classification and evaluate the performance of traditional machine learning methods versus deep learning techniques. Through this process, we found that **Random Forest** and **SVM** models, when combined with **TF-IDF** and **Doc2Vec**, provided the highest accuracy in classifying sentiments from text.

However, there were some limitations in our study. The dataset used was not entirely balanced in terms of emotion representation, which might have affected the results, particularly for less common emotions like fear and disgust. Additionally, the computational resources required for deep learning models like Doc2Vec were higher, limiting their scalability for large datasets.

For future improvements, we recommend:

1. Expanding the dataset to ensure better representation of all emotion categories.
2. Exploring advanced models such as **BERT** and **PhoBERT** for better handling of complex language structures.
3. Further optimization of the preprocessing techniques to handle noisy or incomplete data more efficiently.

Overall, this project provided valuable insights into the effectiveness of traditional and deep learning approaches for sentiment analysis and highlighted areas for future enhancement.

# REFERENCE

1. <https://campus.datacamp.com/courses/feature-engineering-for-nlp-in-python/n-gram-models?ex=6>
2. <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/>
3. <https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/>
4. <https://arxiv.org/abs/2003.00744>
5. <https://www.kaggle.com/code/trnmtin/phobert-classification-for-vietnamese-text>

# 