Shape, rectangle

Description automatically generated

**Ho Chi Minh City, December 2022**

Lecturer: Cao Thi Nhan

Nguyen Ho Duy Tri

Students:

19521720 Tran Trong Kien

19521894 Le Huu Nghia

19522551 Nguyen Trong Xuyen

19522098 Nguyen Tuong Minh Quang

**FINAL PROJECT REPORT**

**DATA MINING**

**TOPIC**

**VIETNAMESE TEXT CLASSIFICATION**

**VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY**

**UNIVERSITY OF INFORMATION TECHNOLOGY**

**DEPARTMENT OF INFORMATION SYSTEM**

# LECTURER’S COMMENTS

**…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**…………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

# TABLE OF CONTENTS

[LECTURER’S COMMENTS 2](#_Toc120818269)

[TABLE OF CONTENTS 3](#_Toc120818270)

[CHAPTER I: TOPIC OVERVIEW 4](#_Toc120818271)

[1. Title 4](#_Toc120818272)

[2. Abstract 4](#_Toc120818273)

[3. Introduction 4](#_Toc120818274)

[CHAPTER II: METHOD 6](#_Toc120818275)

[1. Data collecting 6](#_Toc120818276)

[2. Data preprocessing[1] 6](#_Toc120818277)

[**2.1.** **Remove HTML code** 7](#_Toc120818278)

[**2.2.** **Standardize Vietnamese Unicode** 7](#_Toc120818279)

[**2.3.** **Standardize Vietnamese accent typing (using òa úy instead of oà uý)** 7](#_Toc120818280)

[**2.4.** **Tokenize Vietnamese word** 7](#_Toc120818281)

[**2.5.** **Return text to lowercase** 7](#_Toc120818282)

[**2.6.** **Remove special characters (“.”, “,”, “;”, “)”,…)** 7](#_Toc120818283)

[**2.7.** **Remove Vietnamese stopwords** 8](#_Toc120818284)

[3. Data separating 8](#_Toc120818285)

[4. Data fitting in each models 8](#_Toc120818286)

[4.1. Logistic Regression[8,9] 8](#_Toc120818287)

[4.2. K-Nearest Neighbors (KNN)[7] 10](#_Toc120818288)

[4.3. Decision Tree[3,10] 11](#_Toc120818289)

[4.4. Random Forest[11] 13](#_Toc120818290)

[4.5. AdaBoost[12] 13](#_Toc120818291)

[4.6. Naïve Bayes[5] 14](#_Toc120818292)

[4.7. Support Vector Machine (SVM)[6] 15](#_Toc120818293)

[CHAPTER III: RESULT 17](#_Toc120818294)

[CHAPTER IV: CONCLUSION 20](#_Toc120818295)

[CHAPTER V: REFERENCES 21](#_Toc120818296)

# CHAPTER I: TOPIC OVERVIEW

## Title

VIETNAMESE TEXT CLASSIFICATION

## Abstract

Natural Language Processing (NLP) is a subfield of Linguistics, Computer Science, and Artificial Intelligence [4] – which is one of technology trends in nowadays beside 5G, IoT – Internet of Things, Cloud Computing, Big Data, AR – Augmented Reality, VR – Virtual Reality.

Text classification is a common problem in Natural Language Processing. For Vietnamese text classification, it will be a little different from English text classification. This paper will especially focus on the preprocessing of Vietnamese data to get a good enough classification model.[1]

In this paper, we use texts\_categories data – which is being collected from many Vietnamese online newspapers about 2 topics including sport and injury. From empirical analysis of 07 models that we use in this research, we find that SVM (Support Vector Machine) gives the more accurate for classifying categorical label Vietnamese text than any other model (including Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, AdaBoost, Naïve Bayes) even though others also give very high accuracy.

Keywords: Vietnamese text, Vietnamese text classification, Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, AdaBoost, Naïve Bayes, Support Vector Machine (SVM).

## Introduction

In this practical work, we want to classify Vietnamese text. To do that, we begin with researching about NLP (Natural language processing) and classification. NLP give computers the ability to understand text and spoken words in much the same way human beings can.[2]

Classification predicts categorical class labels, classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data.[3]

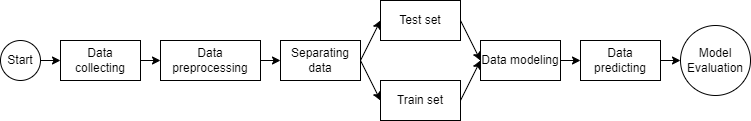
Text Classification is a supervised learning problem in machine learning. This problem requires data to have a label. The model will learn from that labelled data, which is then used to predict labels for new data that the model has not yet encountered. For example, you need to build a machine learning model to predict the topic (like Economics, Social, Sports, ...) of any article. Then you need a lot of labelled data; it means you need a lot of articles, each of which we have to know in advance what topic it is in. The data problem is perhaps the biggest problem of this supervised learning model.[1]

Right here, our research question appear. It is “How best can we classify categorical label of new data?”.

Then we find out that there are 07 classification models suitable to answer our research questions: Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, AdaBoost, Naïve Bayes, Support Vector Machine (SVM).

# CHAPTER II: METHOD

The methodology of a paper is a vital and key feature because this stage discusses the whole process of functional activity. The machine learning algorithm mainly followed the 7 steps of methodology for analyzing the data and predicting the outcome and taking a decision. All steps are data collecting, data preprocessing, separating data for training and testing, data modelling, data predicting (classifying) with testing set and evaluating all the models.

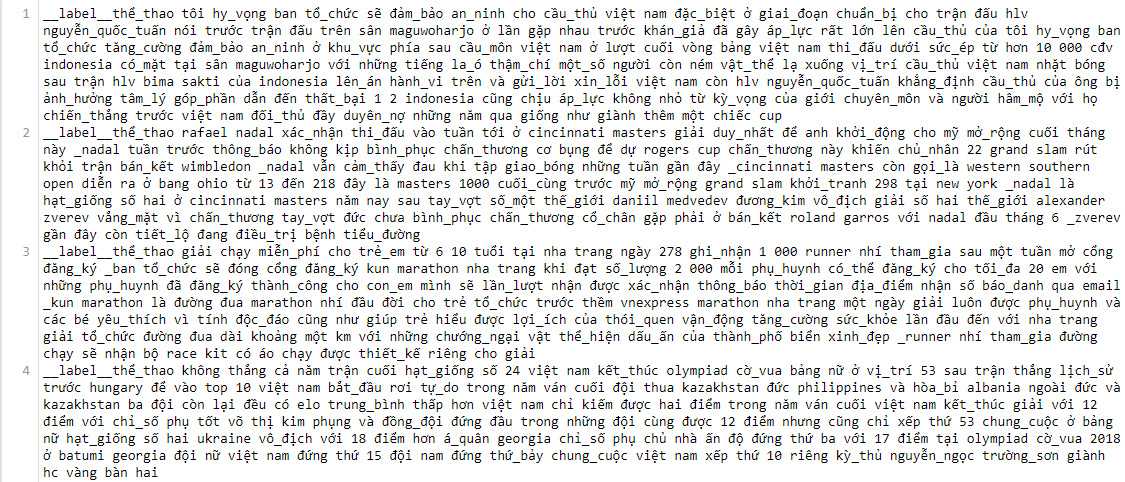


## Data collecting

For the Vietnamese news text classification problem, the data you need to prepare is the data of Vietnamese articles with the topic of that article. This kind of data is generally very easy to find, because there are countless news websites that publish dozens of stories a day.

In this stage, we look at collecting data from many Vietnamese online newspapers about 2 topics including sport and injury. We preprocess one by one text then add it in to our dataset.

Our dataset attributes are categorical label and Vietnamese text. Which is stored in a txt file, which is named texts\_categories. It contains 400 texts equally divided by 02 labels are \_\_label\_\_chấn\_thương and \_\_label\_\_thể thao. The format is:



## Data preprocessing[1]

The data preprocessing step is the first step need to be done. Data preprocessing is the process of normalizing data and removing elements that are not meaningful for text classification.

* 1. **Remove HTML code**

Data collected from websites sometimes still contains HTML snippets. These HTML codes are garbage, not only not effective for classification, but also make the text classification results worse.

* 1. **Standardize Vietnamese Unicode**

Currently, there are two types of Unicode codes in common use, combinatorial Unicode and built-in Unicode. That leads to the problem that even though we are seeing the words as the same but the machine learning model will interpret them as different.

The solution is bring back to built-in Unicode standard (more common).

* 1. **Standardize Vietnamese accent typing (using òa úy instead of oà uý)**

You can see clearly the difference between 2 types of accent typing: òa and oà are respectively the old (more common) and new type of typing.

* 1. **Tokenize Vietnamese word**

The word unit in Vietnamese consists of a single word and a compound word. So we need to tell the machine learning model which is a single word and which is a compound word. Otherwise, every word will be a single word.

Because our model will treat words as features, tokenized by space. Therefore, we have to join the compound words into one word so that there is no wrong tokenization.

This problem is a fundamental problem in NLP - word tokenize problem. Fortunately, there are now quite a few open source libraries for this problem. Therefore, we just need to install and use. With Python language, you can choose underthesea or pyvi.

* 1. **Return text to lowercase**

The inclusion of data in lowercase is essential. Because this feature does not work in the text classification problem. Returning to lowercase reduces the number of features (because computers understand uppercase and lowercase type as two different words) and increases the accuracy of the model.

* 1. **Remove special characters (“.”, “,”, “;”, “)”,…)**

Preprocessing includes removing data that is not useful for text classification. This helps us reduce the number of feature dimensions, speed up learning and processing; avoid adversely affecting the results of the model.

Punctuation marks, numbers, and other special characters don't help you categorize a document by category. Therefore, we should remove it.

* 1. **Remove Vietnamese stopwords**

Stopwords are words that appear a lot in all categories to be classified. Therefore, they are features that are not useful for text classification.

The stopwords are usually the linking words (của, là, có, được, những,…) and data-specific words.

Just like the removal of special characters above, but now Vietnamese words.

* **Set of stopwords**

The set of stopwords should be built from a large text data set. We list the words that appear a lot in all categories and get the top 100 to make a set of stopwords.

Or we can refer to some Vietnamese stopword sets that have been shared on the internet.

* **Remove stopwords from data**

What we need to do now is to go through each data record and delete all the words in the data that are in the stopword set.

## Data separating

The complete data were divided into two separated parts that train and test with the ratio of 80% and 20% respectively. The percentage of trains and tests would impact the accuracy of predicting the result. At this stage what ration you want to choose for the train and test dataset it's up to you but if you take more training set compared to testing set then accuracy would be better. The general ratio for test and train dataset is 80% and 20% respectively.

## Data fitting in each models

Choose an algorithm is vital for getting better predictions and selecting a proper algorithm based on the dataset. For this dataset, we selected Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, AdaBoost, Naïve Bayes, Support Vector Machine (SVM) to fit with the training set and predict with the testing set.

### Logistic Regression[8,9]

Logistic Regression is a statistical model often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as 0 or 1, yes or no, voted or didn’t vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1 [8]. In regression analysis, Logistic Regression is estimating the parameters of a logistic model (the coefficients in the linear combination). Formally, in Logistic Regression there is a single binary dependent variable, coded by an indicator variable, where the two values are labelled "0" and "1", while the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value) [9].

Odds are the probability of an event occurring in a given sample. In a sample of n subjects, and where if we observe that x objects have a characteristic, the probability (P) is defined as:

The definition of odds is defined in terms of P as following:

In Logistic Regression, a logit transformation is applied on the odds - that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas [8]:

P is the probability of the event occurring and X is an independent variable, the Logistic Regression model describes the relationship between X and P as following:

or

In which:

α is the intercept factor when the value X = 0, β is the slope (The beta parameter, or coefficient, in this model is commonly estimated via maximum likelihood estimation (MLE) [8]) and ε is the residuals.

For binary classification, a probability less than 0.5 will predict 0 while a probability greater than 0.5 will predict 1. After the model has been computed, it’s best practice to evaluate how well the model predicts the dependent variable, which is called goodness of fit [8].

### K-Nearest Neighbors (KNN)[7]

In statistics, the K-Nearest Neighbors algorithm (KNN) is a non-parametric supervised learning method first developed by Evelyn Fix and Joseph Hodges in 1951, and later expanded by Thomas Cover. It is used for classification and regression. In both cases, the input consists of the k closest training examples in a data set. The output depends on whether KNN is used for classification or regression:

* In KNN classification, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.
* In KNN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

KNN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically.

Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for KNN classification) or the object property value (for KNN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

A peculiarity of the KNN algorithm is that it is sensitive to the local structure of the data.

The K-Nearest Neighbour classifier can be viewed as assigning the nearest neighbours a weight and all others weight. This can be generalised to weighted nearest neighbour classifiers. That is, where the th nearest neighbour is assigned a weight , with . An analogous result on the strong consistency of weighted nearest neighbour classifiers also holds.

Let denote the weighted nearest classifier with weights . Subject to regularity conditions, which in asymptotic theory are conditional variables which require assumptions to differentiate among parameters with some criteria. On the class distributions the excess risk has the following asymptotic expansion

,

for constants and where and .

The optimal weighting scheme , that balances the two terms in the display above, is given as follows: set ,

for and

for .

With optimal weights the dominant term in the asymptotic expansion of the excess risk is .

### Decision Tree[3,10]

Decision Tree learning is a supervised learning approach used in statistics, data mining and machine learning. In this formalism, a classification or regression decision tree is used as a predictive model to draw conclusions about a set of observations.

Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making).

Tree is contructed in a top-down recursive divide-and-conquer manner. At start, all the training examples are at the root. Attributes are categorical (if continuous-valued, they are discretized in advance). Examples are partitioned recursively based on selected attributes. Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain, Gini index…)

Entropy is measure of uncertainty associated with a random variable. Entropy is used to build the tree. Calculation Entropy of set S:

* S: sample set
* n: number of different values of all samples in S
* Aj: number of sample corresponding to each j
* Fs(Aj): ratio of Aj to S

Information Gain of set of sample S based on attribute A:

* G(S,A): information gain of set S based on attribute A
* E(S): entropy of S
* m: number of different values of attribute A
* Ai: number of sample corresponding to each i of attribute A
* Fs(Ai): ratio of Ai to S
* SAi: subset of S including all samples having value Ai

Gini index of data D:

With: the relative frequency of class in

If a data set is split on A into k subsets the index is defined as:

With:

* ni: #samples of node i
* N: #samples of node A

Select attribute with minimal Gini index for partitioning.

### Random Forest[11]

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.). The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho and later independently by Amit and Geman in order to construct a collection of decision trees with controlled variance.

Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

### AdaBoost[12]

AdaBoost, short for Adaptive Boosting, is a statistical classification meta-algorithm formulated by Yoav Freund and Robert Schapire in 1995, who won the 2003 Gödel Prize for their work. It can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. Usually, AdaBoost is presented for binary classification, although it can be generalized to multiple classes or bounded intervals on the real line.

AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

Although AdaBoost is typically used to combine weak base learners (such as decision stumps), it has been shown that it can also effectively combine strong base learners (such as deep decision trees), producing an even more accurate model.

Every learning algorithm tends to suit some problem types better than others, and typically has many different parameters and configurations to adjust before it achieves optimal performance on a dataset. AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier. When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative 'hardness' of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.

### Naïve Bayes[5]

In statistics, Naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models, but coupled with kernel density estimation, they can achieve high accuracy levels.

Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In the statistics literature, Naïve Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but Naïve Bayes is not (necessarily) a Bayesian method.

* **Multinomial Naïve Bayes**

With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial where is the probability that event i occurs (or K such multinomials in the multiclass case). A feature vector is then a histogram, with counting the number of times event i was observed in a particular instance. This is the event model typically used for document classification, with events representing the occurrence of a word in a single document (see bag of words assumption). The likelihood of observing a histogram x is given by

The Multinomial Naïve Bayes classifier becomes a linear classifier when expressed in log-space:

Where and .

If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero, because the probability estimate is directly proportional to the number of occurrences of a feature's value. This is problematic because it will wipe out all information in the other probabilities when they are multiplied. Therefore, it is often desirable to incorporate a small-sample correction, called pseudocount, in all probability estimates such that no probability is ever set to be exactly zero. This way of regularizing Naïve Bayes is called Laplace smoothing when the pseudocount is one, and Lidstone smoothing in the general case.

Rennie *et al.* discuss problems with the multinomial assumption in the context of document classification and possible ways to alleviate those problems, including the use of TFIDF weights instead of raw term frequencies and document length normalization, to produce a Naïve Bayes classifier that is competitive with Support Vector Machine.

### Support Vector Machine (SVM)[6]

In Machine Learning, Support Vector Machine is supervised learning model with associated learning algorithms that analyze data for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

* **Support Vector Clustering (SVC)**

When data are unlabelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The Support Vector Clustering (SVC) algorithm is a similar method that also builds on kernel functions but is appropriate for unsupervised learning that applies the statistics of support vectors, developed in the Support Vector Machine algorithm, to categorize unlabelled data.

# CHAPTER III: RESULT

The Accuracy score rankings from highest to lowest score are as following:

* 1st: SVM (98.75%);
* 2nd: Logistic Regression (97.5%), Random Forest (97.5%);
* 3rd: KNN (96.25%), Naïve Bayes (96.25%);
* 4th: AdaBoost (91.25%);
* 5th: Decision Tree (88.75%).

Classification Report of Support Vector Machine (SVM):

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| \_\_label\_\_chấn\_thương | 0.97 | 1.00 | 0.99 |
| \_\_label\_\_thể\_thao | 1.00 | 0.98 | 0.99 |

Classification Report of Logistic Regression:

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| \_\_label\_\_chấn\_thương | 0.97 | 0.97 | 0.97 |
| \_\_label\_\_thể\_thao | 0.98 | 0.98 | 0.98 |

Classification Report of Random Forest:

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| \_\_label\_\_chấn\_thương | 0.95 | 1.00 | 0.97 |
| \_\_label\_\_thể\_thao | 1.00 | 0.95 | 0.98 |

Classification Report of K-Nearest Neighbors (KNN):

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| \_\_label\_\_chấn\_thương | 0.92 | 1.00 | 0.96 |
| \_\_label\_\_thể\_thao | 1.00 | 0.93 | 0.96 |

Classification Report of Naïve Bayes:

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| \_\_label\_\_chấn\_thương | 0.92 | 1.00 | 0.96 |
| \_\_label\_\_thể\_thao | 1.00 | 0.93 | 0.96 |

Classification Report of AdaBoost:

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| \_\_label\_\_chấn\_thương | 0.94 | 0.86 | 0.90 |
| \_\_label\_\_thể\_thao | 0.89 | 0.95 | 0.92 |

Classification Report of Decision Tree:

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| \_\_label\_\_chấn\_thương | 0.94 | 0.81 | 0.87 |
| \_\_label\_\_thể\_thao | 0.86 | 0.95 | 0.90 |

For \_\_label\_\_chấn\_thương:

* Precision score:
  + 1st: SVM (0.97), Logistic Regression (0.97);
  + 2nd: Random Forest (0.95);
  + 3rd: AdaBoost (0.94), Decision Tree (0.94);
  + 4th: KNN (0.92), Naïve Bayes (0.92).

Which means SVM, Logistic Regression gives more accurate in predicting true positive points in all points classified into positive group than others.

* Recall score:
  + 1st: SVM (1.00), Random Forest (1.00), KNN (1.00), Naïve Bayes (1.00);
  + 2nd: Logistic Regression (0.97);
  + 3rd: AdaBoost (0.86);
  + 4th: Decision Tree (0.81).

Which means SVM, Random Forest, KNN, Naïve Bayes gives absolutely accurate in predicting true positive points in all actual positive points, better than others.

* F1 Score:
  + 1st: SVM (0.99);
  + 2nd: Logistic Regression (0.97), Random Forest (0.97);
  + 3rd: KNN (0.96), Naïve Bayes (0.96);
  + 4th: AdaBoost (0.90);
  + 5th: Decision Tree (0.87).

F1 Score rankings is the same as Accuracy rankings in every ranking.

For \_\_label\_\_thể\_thao:

* Precision score:
  + 1st: SVM (1.00) , Random Forest (1.00), KNN (1.00), Naïve Bayes (1.00);
  + 2nd: Logistic Regression (0.98);
  + 3rd: AdaBoost (0.89);
  + 4th: Decision Tree (0.86).

Which means SVM, Random Forest, KNN, Naïve Bayes gives absolutely accurate in predicting true positive points in all points classified into positive group, better than others.

* Recall score:
  + 1st: SVM (0.98), Logistic Regression (0.98);
  + 2nd: Random Forest (0.95), AdaBoost (0.95), Decision Tree (0.95);
  + 3rd: KNN (0.93), Naïve Bayes (0.93).

Which means SVM, Logistic Regression gives more accurate in predicting true positive points in all actual positive points than others.

* F1 Score:
  + 1st: SVM (0.99);
  + 2nd: Logistic Regression (0.98), Random Forest (0.98);
  + 3rd: KNN (0.96), Naïve Bayes (0.96);
  + 4th: AdaBoost (0.92);
  + 5th: Decision Tree (0.90).

F1 Score rankings is the same as Accuracy rankings in every ranking.

Predict categorical label for new data:

|  |  |
| --- | --- |
| “Trong thể thao, chấn thương là chuyện không thể tránh khỏi của vận động viên” | |
| Support Vector Machine (SVM) | \_\_label\_\_chấn\_thương |
| Logistic Regression | \_\_label\_\_chấn\_thương |
| Random Forest | \_\_label\_\_chấn\_thương |
| K-Nearest Neighbors (KNN) | \_\_label\_\_chấn\_thương |
| Naïve Bayes | \_\_label\_\_chấn\_thương |
| AdaBoost | \_\_label\_\_chấn\_thương |
| Decision Tree | \_\_label\_\_chấn\_thương |

|  |  |
| --- | --- |
| “Nghĩa đá bóng và chạy bộ rất thường xuyên” | |
| Support Vector Machine (SVM) | \_\_label\_\_thể\_thao |
| Logistic Regression | \_\_label\_\_thể\_thao |
| Random Forest | \_\_label\_\_thể\_thao |
| K-Nearest Neighbors (KNN) | \_\_label\_\_chấn\_thương |
| Naïve Bayes | \_\_label\_\_thể\_thao |
| AdaBoost | \_\_label\_\_thể\_thao |
| Decision Tree | \_\_label\_\_thể\_thao |

|  |  |
| --- | --- |
| “Nghĩa bị chấn thương khi tập luyện thể thao quá mức” | |
| Support Vector Machine (SVM) | \_\_label\_\_chấn\_thương |
| Logistic Regression | \_\_label\_\_chấn\_thương |
| Random Forest | \_\_label\_\_chấn\_thương |
| K-Nearest Neighbors (KNN) | \_\_label\_\_chấn\_thương |
| Naïve Bayes | \_\_label\_\_chấn\_thương |
| AdaBoost | \_\_label\_\_chấn\_thương |
| Decision Tree | \_\_label\_\_chấn\_thương |

As the above figure shows that we predict categorical label for 3 new texts. All models give correct results which means the predicted labels are the same as our expected labels except KNN gives incorrect result in the second new text.

# CHAPTER IV: CONCLUSION

Machine learning has some great applications and still, now it's a very popular tool and it's also depending much on data even though it has evolved the future into AI and deep learning. All about this paper is classifying categorical label for Vietnamese text using machine learning models and algorithms. There are various ways to implement the Vietnamese text classification but we use 07 models to study in this investigation. The main aim of this paper is to use supervised machine learning algorithms like Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, AdaBoost, Naïve Bayes, Support Vector Machine (SVM) for Vietnamese text classification. The result reveals that SVM (Support Vector Machine) gives the more accurate for classifying categorical label Vietnamese texts than any other model (including Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), Naïve Bayes, AdaBoost, Decision Tree) even though others also give very high accuracy.

The model rankings for Vietnamese text classification (subjective opinion based on empirical analysis):

* 1st: Support Vector Machine (SVM);
* 2nd: Logistic Regression, Random Forest;
* 3rd: K-Nearest Neighbors (KNN), Naïve Bayes;
* 4th: AdaBoost;
* 5th: Decision Tree.

In our experience, F1 Score will give low score if the value of Precision or Recall is low. F1 Score requires both Precision and Recall to get high rate. So that it’s suitable for evaluating the overview of the model much more better than just Accuracy score, Precision or Recall.

Through this study, our target is building a foundation for automating Vietnamese text classification. After this research, we will continue to build a larger dataset with more than just 02 labels and find a way to automate this laborious process, also try to keep (even improve) the excellent accuracy. Furthermore is to build an AI that can do all these things itself.

# CHAPTER V: REFERENCES

[1] Nguyen Van Hieu, “Phân loại văn bản tiếng Việt sử dụng machine learning”, [link: <https://blog.luyencode.net/phan-loai-van-ban-tieng-viet/>].

[2] IBM Cloud Education, “Natural Language Processing (NLP)”, [link: <https://www.ibm.com/cloud/learn/natural-language-processing#toc-what-is-na-jLju4DjE>], July 2, 2020.

[3] Cao Thi Nhan, “Chapter 6 Classification”, Nhan\_Chapter 6\_Classification 2022.pdf.

[4] Wikipedia, “Natural language processing”, [link: <https://en.wikipedia.org/wiki/Natural_language_processing>].

[5] Wikipedia, “Naive Bayes classifier”, [link: <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>].

[6] Wikipedia, “Support vector machine”, [link: <https://en.wikipedia.org/wiki/Support_vector_machine>].

[7] Wikipedia, “k-nearest neighbors algorithm”, [link: <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>].

[8] IBM, “What is logistic regression?”, [link: <https://www.ibm.com/topics/logistic-regression>].

[9] Wikipedia, “Logistic regression”, [link: <https://en.wikipedia.org/wiki/Logistic_regression#Parameter_estimation>].

[10] Wikipedia, “Decision tree learning”, [link: <https://en.wikipedia.org/wiki/Decision_tree_learning>].

[11] Wikipedia, “Random forest”, [link: <https://en.wikipedia.org/wiki/Random_forest>].

[12] Wikipedia, “AdaBoost”, [link: <https://en.wikipedia.org/wiki/AdaBoost>].