VNUHCM - UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF KNOWLEDGE ENGINEERING

INTRODUCTION TO NATURAL LANGUAGE PROCESSING

PROJECT 02:

TEXT CLASSIFICATION

GVHD:	Dr. Nguyen Hong Buu Long
GVTH:	Dr. Le Thanh Tung
	Dr. Luong An Vinh

HO CHI MINH CITY
OCTOBER/2023

Contents

1 Introduction

1.1 Text classification

Text classification, often referred to as text categorization or document classification, is a fundamental task in natural language processing (NLP) and machine learning. It involves the process of automatically assigning predefined categories or labels to a given piece of text based on its content. The primary objective of text classification is to make sense of unstructured textual data by organizing it into meaningful and manageable groups. This task has numerous real-world applications, ranging from spam email detection and sentiment analysis to content recommendation systems and information retrieval.

Text classification algorithms use machine learning techniques to analyze the features and patterns within a text document to determine its category. These algorithms typically rely on a labeled dataset for training, where texts are associated with their respective categories. Once trained, the model can generalize its learning to classify new, unseen documents accurately.

The applications of text classification are diverse and continue to expand, driven by the exponential growth of digital textual data. It is an essential component of various NLP tasks and information management systems, playing a crucial role in automating and enhancing the efficiency of content processing and decision-making in a wide range of industries, including e-commerce, healthcare, finance, and social media analysis.

1.2 Scikit-learn

Scikit-learn, also known as sklearn, is a popular open-source machine learning library for Python. It provides a wide range of tools and algorithms for various tasks such as classification, regression, clustering, and dimensionality reduction. Scikit-learn is built on NumPy, SciPy, and matplotlib, which are other powerful libraries for scientific computing and data visualization in Python.

One of the key strengths of scikit-learn is its simplicity and ease of use. It offers a consistent and intuitive API, making it relatively straightforward for both beginners and experienced practitioners to work with. Scikit-learn also emphasizes code readability and maintainability, making it an excellent choice for building machine learning models in production environments.

The library provides a comprehensive set of functionalities for data preprocessing, feature engineering, model selection, and evaluation. It includes a wide range of algorithms such as support vector machines (SVM), random forests, gradient boosting, k-means clustering, and many others. These algorithms are implemented efficiently and optimized for performance, allowing users to work with large datasets and complex models.

Scikit-learn is widely used in both academia and industry due to its versatility and robustness. It has a large and active community, which contributes to its continuous development and improvement. Additionally, scikit-learn integrates well with other libraries in the Python ecosystem, making it an essential tool for data scientists and machine learning practitioners.

Whether you're a beginner exploring machine learning concepts or an experienced practitioner deploying models in real-world applications, scikit-learn provides a powerful and user-friendly framework to support your machine learning endeavors.

2 Project Requirement

To successfully complete this project, you will need to utilize Python (version 3.0) and Scikit-learn on the Google Colab platform. You can enhance your understanding of Scikit-learn by referring to the following websites:

Scikit-learn Tutorials: You can access a comprehensive tutorial on Scikit-learn at the official documentation website. Simply visit https://scikit-learn.org/stable/tutorial/index.html to explore the tutorial and learn more about the library.

DataCamp Scikit-Learn Tutorial: Another valuable resource is the Scikit-Learn tutorial provided by DataCamp. Visit https://www.datacamp.com/tutorial/machine-learning-python to access this tutorial. It offers an easy-to-follow guide on Scikit-Learn, helping you get started with Python machine learning.

By utilizing these resources, you will gain a solid foundation in Scikit-learn and be equipped with the knowledge necessary to tackle your project effectively.

2.1 Theory Part

The students are required to research and answer the following questions, accompanied by corresponding examples in English. Try to provide clear and concise responses.

1. What are the differences between supervised, unsupervised, and reinforcement learning, and how do they differ in their learning approaches?

2. When evaluating the performance of a text classification model, what are the common evaluation metrics used, and how can we assess its effectiveness?

3. In the context of text classification, what are some techniques that can be employed to handle imbalanced datasets and address the challenges associated with class imbalance?

4. Support Vector Machines (SVMs) are primarily designed for binary classification. Can you explain the approach used to extend SVMs for handling multi-class classification problems and how it enables them to classify data into multiple classes?

5. How would you define the kernel within the SVM algorithm? Furthermore, could you introduce a few kernel functions offered in scikit-learn and elaborate on the variations among them?

2.2 Programming

This is a group project. Each group will consist of 3-4 members (In case you cannot find a group, please fill in the following link: Click here, and you will be randomly assigned to a group with similar students).

Each group is required to perform the following tasks using Python on the Colab platform. The tasks include:

1. Conduct research and implement the algorithms: Decision Tree, Random Forest, SVM, CRF for sentence classification.

2. Investigate the parameters that impact the system's quality and training process (details provided in the attached file).

3. Build evaluation charts and present the results automatically.

3 Submission

The submission file must be in the following format: [**StudentID1_StudentID2_StudentID3.zip**], is the compression of the [**StudentID1_StudentID2_StudentID3**] folder. The order of student IDs is sorted in ascending.

This folder contains:

- The [**StudentID1_StudentID2_StudentID3.ipynb**] downloaded from Google Colab. Before submitting, please choose **"Run All"** and make sure that your file can run normally
- In the Python Notebook, please add a text box and insert the information including your name and your student ID.
- The [**StudentID1_StudentID2_StudentID3.pdf**] and [**StudentID1_StudentID2_StudentID3.docx**] contains the main report to answer all theory questions and present the discussion in Programming parts. The requirements of this report is shown in the attached file.
- The theory part should be **structured, logical, clear** and **coherent**.
- All links and books related to your submission must be mentioned.