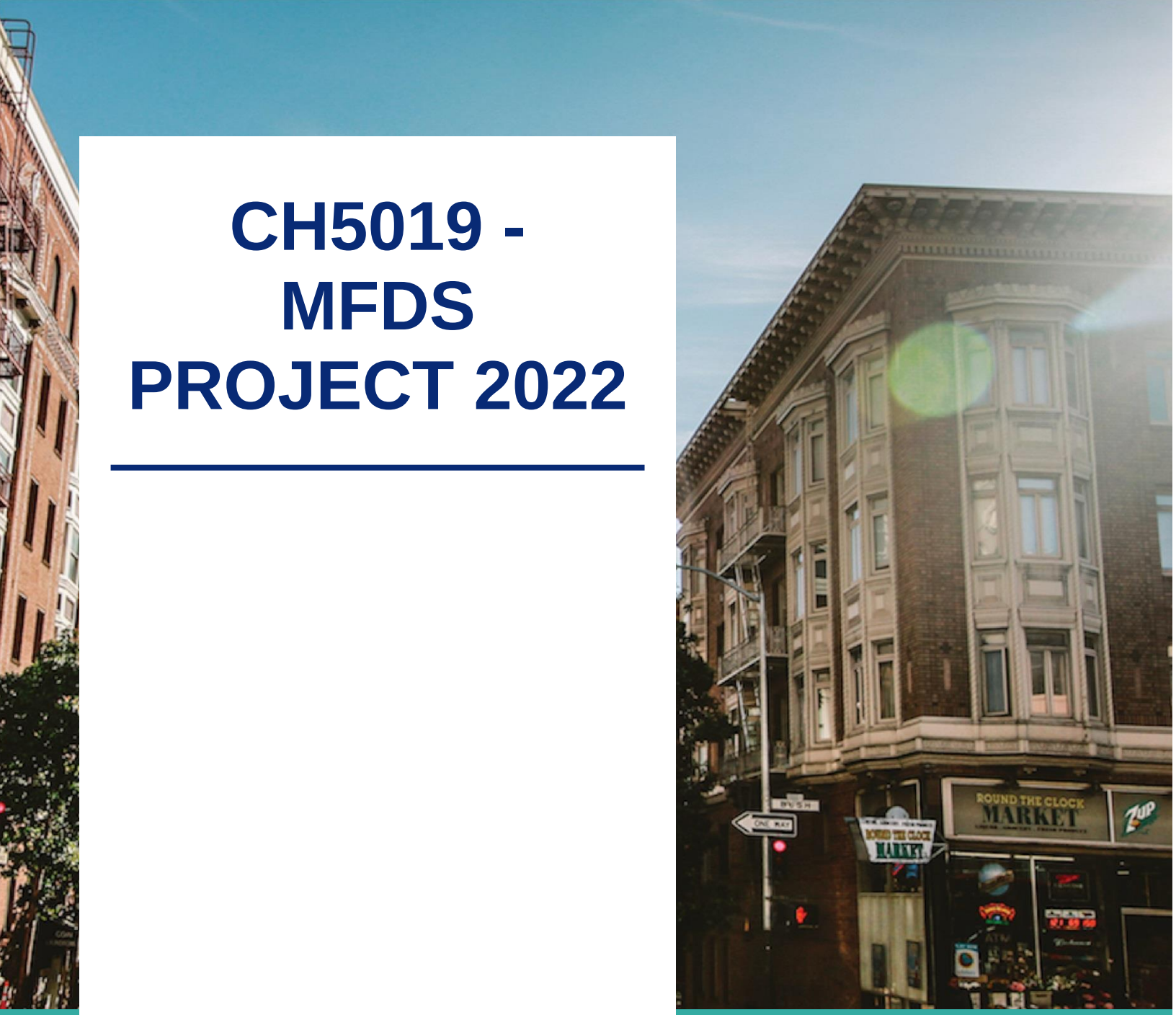

CH5019 - MFDS PROJECT 2022

GROUP 18

CH19B059 – KJLDK Srija

CH19B080 – Pruthvi Akare

CH19B091 – D V Sridurgaa

CH5019 - Mathematical Foundations of Data Science

Course Project

An instructor wants to grade answers to descriptive questions automatically. The instructor has a template best answer that she/he has developed and wants to use the same to grade descriptive answers of students. The comparison results between the template and the student provided answer could be categorical (right vs wrong) or continuous (75% similarity and so on). You are expected to develop an AI algorithm that can do this comparison automatically. Please develop an algorithm through any appropriate concept and demonstrate the results on ten test cases. You can use any training approach and test it on any 10 test cases that you feel are appropriate. Your algorithm should take two paragraphs, one correct answer and one student provided answer and return a result. Since this is a rather open-ended problem, the solutions will be largely evaluated based on the creativity associated with problem formulation, solution approach (including feature engineering), and the achieved results. Please remember that there is no single correct method. Please use existing AI/ML libraries as much as possible.

Instructions:

- Groups up to a maximum of five members can be made.
- Code that can be run by the TAs on our own test cases should be submitted.
- A detailed report on the method used, training approach and test cases should be submitted. The maximum page limit for this is 10.
- A contribution statement that clearly articulates the contribution of each group member is mandatory at the end of the report. Reports without contribution statement will not be graded.

Report

It's a complex process to turn text into something that an algorithm can understand. This report will discuss the steps involved in it.

Step 1: Data Preprocessing

- Tokenization is the process of converting sentences into words.
- Tags, removing superfluous punctuation
- Getting rid of stop words, which are common terms like "the," "is," and others that don't have a distinct semantic meaning.
- Words are reduced to a root by eliminating extraneous characters, generally a suffix, and removing inflection.
- Lemmatization — Determining the part of speech and utilizing a detailed database of the language is another method for removing inflection.

Stemming & lemmatization help reduce words like ‘studies’, ‘studying’ to a common base form or root word ‘study’. We can use python to do many text pre-processing operations line NLTK - The Natural Language Tool Kit is one of the best-known and most-used NLP libraries, useful for all sorts of tasks from t tokenization, stemming, tagging, parsing, and beyond.

Step 2: Feature Extraction

In text processing, words of the text represent discrete, categorical features. To represent such data in a way that the algorithms can understand, Feature extraction, the process of converting textual input into real-valued vectors is very useful.

Step 3: Choosing ML Algorithms

#define repository

model = SentenceTransformer('sentence-transformers/bert-base-nli-mean-tokens')

The model sentence transformers bert base-nli-mean-tokens is a Natural Language Processing (NLP) Model implemented in Transformer library, generally using the Python programming language. It is easily in transformers python library. Sentence Embedding using

Siamese BERT-Networks. The sentence-transformers repository allows to train and use Transformer models for generating sentence and text embedding. Sentence Transformers is a Python framework for state-of-the-art sentence, text and image embeddings. The initial work is described template.
The encoder is built with an Embedding layer that converts the words into a vector and a recurrent neural network (RNN)

embedd1 = model.encode(sentence1)
unit_vector1 = embedd1/np.linalg.norm(embedd1)

numpy. linalg. norm is used to calculate the norm of a vector or a matrix.
The dot product of two arrays is

var = np.dot(unit_vector_1,unit_vector_2)

The process of developing morphological variants of a root/base word is known as stemming. Stemming algorithms or stemmers are terms used to describe stemming programmes. A stemming algorithm reduces the words “chocolates”, “chocolatey”, “choco” to the root word, “chocolate”.

import nltk
from nltk.stem import PorterStemmer
ps = PorterStemmer()
print(ps.stem(sentence1))

punkt is the required package for tokenization. Hence you must download it using nltk download manager or download it programmatically using nltk. download('punkt')

nltk.download("punkt")

Tokenization is the process of breaking down a given text into individual words in Natural Language Processing. If a text input document contains paragraphs, it can be split down into

sentences or words. Tokenization is provided by NLTK at two levels: word level and sentence level. The word tokenize() function in NLTK can be used to tokenize a given text into words. You can also use the sent tokenize() function to tokenize provided text into phrases.

word_tokens = nltk.word_tokenize(sentence)
sent_tokens = nltk.sent_tokenize(sentence)

The np.linalg.norm () method takes arr, ord, axis, and keepdims as arguments and returns the norm of the given matrix or vector. The numpy.linalg.norm () method is used to get one of eight different matrix norms or vector norms. The sim refers similarity between two sentences

sim = np.dot(e1,e2)/(np.linalg.norm(e1)*np.linalg.norm(e2))

CONTRIBUTION

Roll No.	Work	Percentage Contribution
CH19B059	Coding and help documentation	33.33%
CH19B080	Explaining algorithm and code in words	33.33%
CH19B091	Coding and help in explaining functions	33.33%