


# COMP 2211 Exploring Artificial Intelligence


## Programming Assignment 1 K-Nearest Neighbors Classifier for Sentiment Analysis

### Sentiment Analysis




My experience so far has been fantastic!

POSITIVE



The product is ok I guess

NEUTRAL



Your support team is useless

NEGATIVE

### Introduction

#### Text Classification

Text classification is a machine learning technique that assigns a set of pre-defined classes to 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.

Text classification is an important and fundamental task with broad applications. For example, it can be used in spam detection (to classify whether a message or email is spam), news categorization (to classify the vast growing online news into some topics), and sentiment analysis (to classify whether a product review is positive, negative or neutral).

You can try the [Google Cloud NLP](#) API to perform the text classification. Type a paragraph in **Try the API** box and get the classification results under the **Categories** button.

### Menu

- [Introduction](#)
- [Assignment Tasks](#)
- [Grading Scheme](#)
- [Submission & Deadline](#)
- [Frequently Asked Questions](#)

### Page maintained by

Dr. XIAO Huiru (Pearl)  
Email: [huiruxiao@ust.hk](mailto:huiruxiao@ust.hk)  
Last Modified:  
02/28/2023 15:52:51

### Homepage

[Course Homepage](#)

DEMO

# Natural Language API demo

Try the API

Try the API

Text classification is a machine learning technique that assigns a set of pre-defined classes to 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.

[See supported languages](#)

EntitiesSentimentSyntaxCategories

/Science/Computer Science  
Confidence: 0.74

/Jobs & Education/Education  
Confidence: 0.5

/Computers & Electronics  
Confidence: 0.63

[See a complete list of content categories.](#)

## Sentiment Analysis

The sentiment analysis task is one kind of text classification task where the algorithm needs to determine whether data are positive, negative or neutral (sometimes the classification is even more fine-grained that there may be more than 3 sentiment labels). Sentiment analysis is often performed to help buisness monitor brand and product sentiment in customer feedback, and understand customer needs.

You can try the [free sentiment analysis demo](#) to perform the sentiment analysis. Type a product review in the box and get the analysis results by clicking the **RUN ANALYSIS** button.

Please enter your text in **english\*** for analysis or leave default one.

I purchased a larger one for bedroom and it arrived with a busted screen, so I ordered a replacement and got it on Friday. Took it out and set it up. NO picture - only static with a BLACK screen. It was hooked to Direct TV so we knew there was a problem when there was no picture and only static. It wouldn't respond to remote buttons or the buttons on the TV itself - definitely a problem. I called LG customer service and we performed a couple of their tests recommendations

☐ Twitter-like content ⓘ

SHARE THIS ANALYSIS

RUN ANALYSIS

I purchased a larger one for bedroom and it arrived with a busted screen, so I ordered a replacement and got it on Friday. Took it out and set it up. NO picture - only static with a BLACK screen. It was hooked to direct TV so we knew there was a problem when there was no picture and only static. It wouldn't respond to remote buttons or the buttons on the TV itself - definitely a problem. I called LG customer service and we performed a couple of their tests recommendations and finally got voice sound but still no picture, then we lost the voice again. The Customer Service lady told me this LG was defective. VERY disappointing to say the least - to receive not one but 2 broke/defective TV's. I'm ready to get my money back and try another brand. And I really do like my smaller LG so this is even more upsetting!

there was no picture broke defective tv's lost the voice busted screen static least defective busted disappointing

This document is: negative (-0.62) ⓘ Magnitude: 5.60

Subjectivity: subjective

Score Range

negativeneutralpositive

-1-0.25+0.25+1

In this assignment, you will build a K-Nearest Neighbors classifier for sentiment analysis.

End of Introduction

## Assignment Tasks

The following tasks are given to you to practice your skills in building an AI model using the K-Nearest Neighbors Classifier.


- Task 1: K-Nearest Neighbors Classifier

https://course.cse.ust.hk/comp2211/assignments/pa1/

2/6

- Task 1.1: Compute the Distance Metric
  - Task 1.2: Find K-Nearest Neighbor Labels
  - Task 1.3: Predict Label for Test Data
- Task 2: Evaluation Metrics
  - Task 2.1: Confusion Matrix
  - Task 2.2: Accuracy
  - Task 2.3: Macro Average Precision
  - Task 2.4: Macro Average Recall
  - Task 2.5: Macro-F1 Score
  - Task 2.6: Matthews Correlation Coefficient
- Task 3: D-Fold Cross-Validation
  - Task 3.1: Generate Folds
  - Task 3.2: Cross-Validate
  - Task 3.3: Validate Best Parameters

Please download the [notebook](#), the [Twitter US Airline Sentiment data](#), [training dataset](#), [training class labels](#), [test dataset](#), and [test class labels](#) and open the notebook using Google Colab. You should see the following if you open the notebook successfully.

 pa1\_sentiment\_analysis\_knn.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Setup: Import the Libraries

Run this code cell to import the libraries that you may need to use in the task. You may also import other modules, as long as they are part of the Python Standard Library. <https://docs.python.org/3/library/>

You are **NOT** allowed to import any other external libraries (e.g., sklearn).

```
[ ] import numpy as np
import pandas as pd
import scipy.sparse as sparse
from scipy import stats
```

End of Assignment Tasks

## Grading Scheme

The grading scheme is as follows (15 points total):

- `compute_Minkowski_distance()` correctly computes the Minkowski distance between the training and test datasets. (1.5 points)
- `find_k_nearest_neighbor_labels()` correctly finds the K nearest neighbor labels for the test dataset. (1.5 points)
- `predict()` correctly returns the predicted classes for the test dataset. (1.5 points)
- `generate_confusion_matrix()` returns the correct confusion matrix given arbitrary test\_predict and test\_label. (1.5 points)
- `calculate_accuracy()` calculates the correct accuracy score given arbitrary test\_predict and test\_label. (1 point)
- `calculate_precision()` calculates the correct macro average precision score given arbitrary test\_predict and test\_label. (1 point)
- `calculate_recall()` calculates the correct macro average recall score given arbitrary test\_predict and test\_label. (1 point)
- `calculate_macro_f1()` calculates the correct macro F1 score given arbitrary test\_predict and test\_label. (1 point)
- `calculate_MCC_score()` calculates the correct Matthews Correlation Coefficient score given arbitrary test\_predict and test\_label. (1.5 points)
- `generate_folds()` correctly generate the training D folds and the testing D folds. (1 point)
- `cross_validate()` correctly calculates the evaluation metric scores. (1.5 points)
- `validate_best_parameters()` correctly returns the best parameters according to the scores. (1 point)

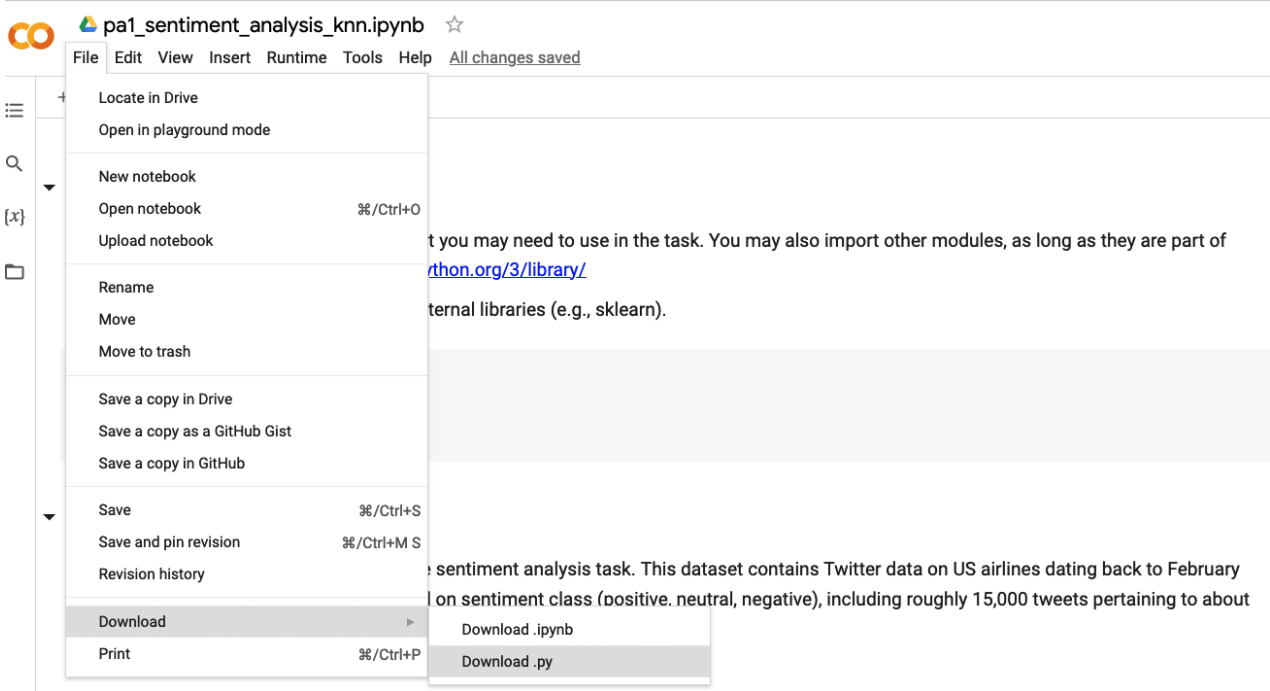
# Submission and Deadline

Deadline: 23:59:00 on 25 March, 2023 (Saturday).

## ZINC Submission

Create a single zip file that contains `pa1_sentiment_analysis_knn.py` file, only this file, NOT a folder containing them. You can create your zip file by the following:

- 1. Get the .py file from Google Colab by clicking File -> Download -> Download .py.



- 2. Compress the python file to `pa1_sentiment_analysis_knn.py.zip`.
- 3. Submit the zip file to [ZINC](#). ZINC usage instructions can be found [here](#).

### Notes:

- You may submit your file multiple times, but only the latest version will be graded.
- Submit early to avoid any last-minute problems. Only ZINC submissions will be accepted.
- This semester we won't grade students' labs/assignments submitted before the deadline in real-time. We will use ZINC to grade your submissions after the submission deadline. You can appeal based on the grading reports in ZINC after the grading is released.

## Requirement

It is **required** that your submissions can be run successfully in our online autog-rader ZINC. It won't be graded if we cannot run your work. Therefore, for parts you cannot finish, just put in a dummy implementation so that your whole program can be run for ZINC to grade the other parts you have done. Empty implementations can be like:

```
def SomeFunctionICannotFinishRightNow():  
    return 0
```

## Reminders

**Make sure you actually upload the correct version of your source files - we only grade what you upload.** Some students in the past submitted an empty file or a wrong file worth zero marks. So **you must double-check the file you have submitted.**

## Late Submission Policy

There will be a penalty of -1 point (out of a maximum of 100 points) for every minute you are late. For instance, since the deadline for assignment 1 is 23:59:00 on March 25th, submitting your solution at 1:00:00 on March 26th will be a penalty of -61 points for your assignment. However, the lowest grade you may get from an assignment is zero: any negative score after the deduction due to a late penalty (and any other penalties) will be reset to zero.

End of Submission and Deadline

## Frequently Asked Questions

**Q:** Will there be any limitation on the maximum time of execution regarding to PA1, or any other programming tasks?

**A:** Yes, there will be execution time limit in ZINC for each lab/pa since the ZINC server cannot always wait for the outputs. Since in PA1, different tasks's computation costs vary a lot, the time limits will be different for each task. We are still testing the appropriate time limits and will update it in PA1 webpage. However, the time limit is only for your reference, even if your code can run within the time limit in google colab or your own environment, it can not be guaranteed that your code can pass the time limit in ZINC, because the environment in ZINC may be different from yours, and the running time in ZINC includes all previous steps of the tested function, for example, loading datasets, running your previous functions that are necessary for the tested function. Don't worry, we will set the time limit in ZINC long enough for the previous steps.

If you utilize what have been taught in class and in lab sessions and try to use numpy vectorization instead of too many loops, your solution will always run within reasonable amount of time. Even if you use loops, your solution will pass the majority of the test cases. But we will set a few (only a few) "stress" test cases that involve larger datasets to test whether your functions are efficient.

**Q:** Will the input dataset always be 2D array and the input labels always be 1D?

**A:** Yes. The input array's dimensions will remain the same even when we use other datasets to grade the submissions, but the input array's shapes may change. For example, for KNNModel, the train\_dataset will be a Numpy 2D array with shape (num\_train\_samples, dimension) and the train\_labels will be a Numpy 1D array with shape (num\_train\_samples, ), but their shapes won't be the same with the notebook ((900, 2642) and (900, )), since the number of data samples, data's dimension (i.e., the number of features), and the number of classes may change.

**Q:** Having a tie for KNN? It seems that in the first test data with self.k = 10, it will have the count of 0 and the count of 1 be equal to 4 (In exact, four 0s, four 1s, and two 2s), how should I deal with this?

**A:** For the sake of simplicity, if there is a tie, please break the tie by following the numerical order, i.e., choosing the smallest index that has the maximum count. For example, if the counts of class 0 and class 1 are both 4, then the prediction is class 0. By the way, in lab3, how to break a tie in KNN classifier will be covered. Welcome to attend lab3 to know more about breaking a tie.

**Q:** My code doesn't work. There is an error/bug. Here is the code. Can you help me fix it?

**A:** As the assignment is a major course assessment, to be fair, you are supposed to work on it on your own, and we should not finish the tasks for you. We are happy to help with explanations and advice, but we shall not directly debug the code for you.

**Q:** Are we allowed to use external libraries (e.g., `scikit-learn`) to implement this assignment?

**A:** In this assignment, you are **NOT** allowed to import extra external libraries (i.e., no `scikit-learn`). This assignment aims to implement a K-Nearest Neighbors classifier from scratch and to give you a hands-on experience of what is going on underneath `scikit-learn`.

**Q:** Are we allowed to use Python standard libraries (e.g., `from collections import defaultdict`)?

**A:** Yes, Python standard libraries are allowed. Please visit [here](#) for an official comprehensive list of modules included in Python 3.

**Q:** 401 Unauthorised Error in PA1 Setup.  
**A:** Make sure you use the same username/password as the one you use to log in to the course website.

End of Frequently Asked Questions

Maintained by COMP 2211 Teaching Team © 2023 HKUST Computer Science and Engineering