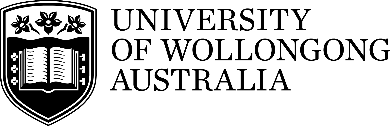
**SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**

**  
INDIVIDUAL Assignment Coversheet**

*This form is to be completed by students submitting* ***online copies*** *of essays or assignments for a Faculty of the Arts, Social Sciences and Humanities subject for the School of Geography and Sustainable Communities, School of Education, and School of Health and Society.*

**PLAGIARISM**

**Deliberate plagiarism may lead to failure in the subject.** Plagiarism is cheating by using the written ideas or submitted work of someone else. The University of Wollongong has a strong policy against plagiarism. See Acknowledgement Practice/Plagiarism Prevention Policy at [**http://www.uow.edu.au/about/policy/UOW058648.html**](http://www.uow.edu.au/about/policy/UOW058648.html)

**Student Name: TEY JIN WAN 7-digit UOW ID: 7435903**

**Subject Code & Name: CSCI218**

**Assignment Title: ASSESSED LAB 1 and 2 (NLP and Search Algo)**

**Tutorial Group: TP01**

(T02, T03, T04, T05)

**Tutor’s Name: Sionggo Japit**

**Assignment Due Date: 24TH FEB 2025**

**DECLARATION**

I certify that this is entirely my own work, except where we have given fully documented references to the work of others, and that the material contained in this assignment has not previously been submitted for assessment in any formal course of study. I understand the definition and consequences of plagiarism.

**ACKNOWLEDGEMENT**

The marker of this assignment may, for the purpose of assessing this assignment, reproduce this assignment and provide a copy to another member of academic staff. If required to do so, we will provide an electronic copy of this assignment to the marker and acknowledge that the assessor of this assignment may, for the purpose of assessing this assignment:

a) Reproduce this assignment and provide a copy to another member of academic staff; and/or

b) Communicate a copy of this assignment to a plagiarism checking service such as Turnitin (which may then retain a copy of this assignment on its database for the purpose of future plagiarism checking).

**Student Signature: TEY JIN WAN Date: 20 Feb 2025**

## Assessed Lab 1: NLP

(**Label CLEARLY your answer to each question**)

### Answers:

1. **Describe the key steps for data preparation and feature extraction (1 mark).**

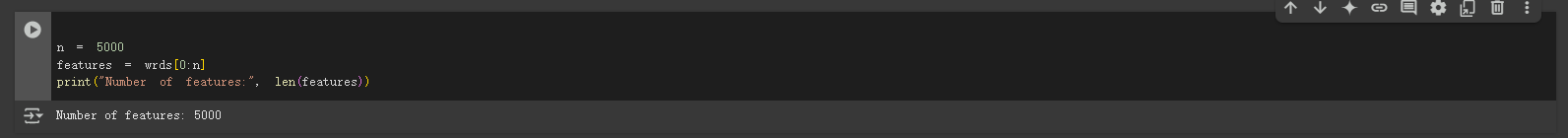
The following key steps have been completed in the data preparation and feature extraction code:

**Step 1:** Data set loading: Loads all document paths and labels from the 20\_newsgroups folder.

**Step 2:** Split the dataset: Use train\_test\_split to split the data set by 75% training set and 25% test set.

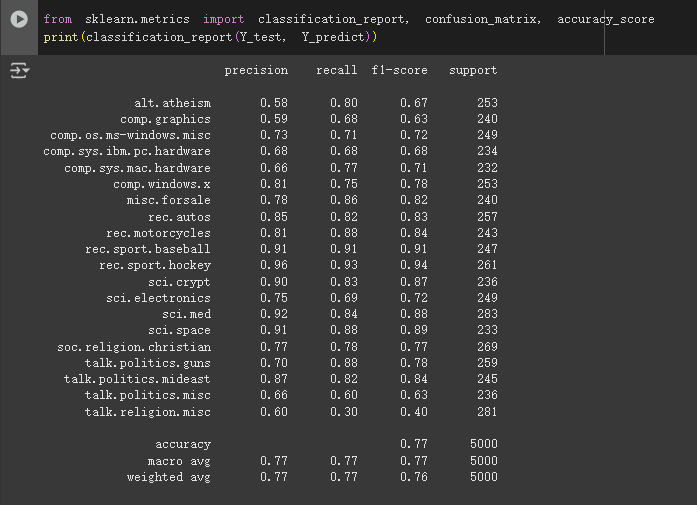
**Step 3:** Text preprocessing: Remove punctuation (keep single quotes), stop words, numbers, single characters, and short words. Unify words in lower case.

**Step 4:** Feature extraction: Count the frequencies of all words in the training set, and select the first 5000 high-frequency words as features. Each document is represented as a word frequency vector (BOW model).



1. **Report the overall classification results, including precision, recall, and f1-score.Explain the meaning of these criteria (1 mark).**

Classification results and index interpretation. After running MultinomialNB, the output classification report is as follows:



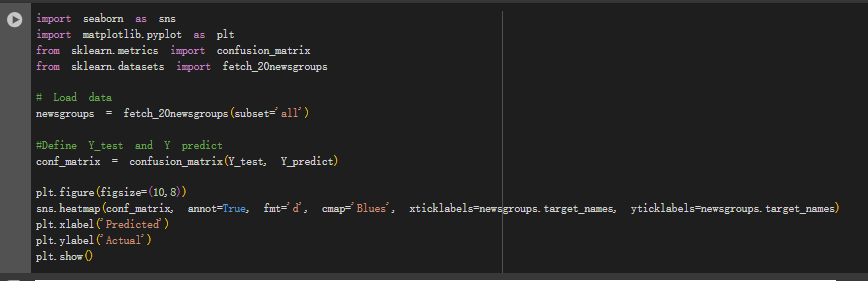
a) Precision: The proportion of a sample that is predicted to be positive is actually positive.

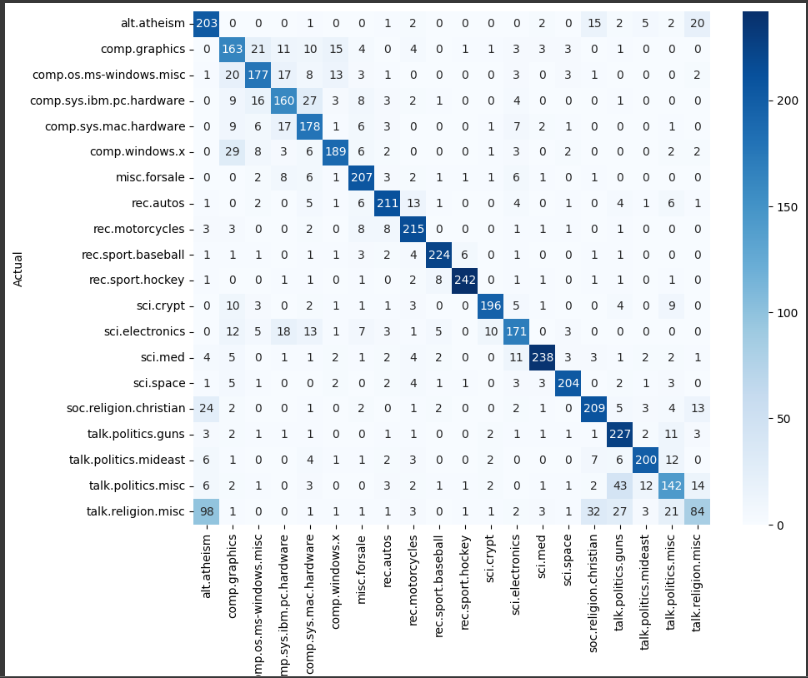
b) Recall: The proportion of a true positive sample that is correctly predicted.

c) F1 Score: A harmonic average of accuracy and recall, balancing the two.

1. **Plot the confusion matrix for your classification result. Find the pair of classes that confuses the classifier most. Is this result consistent with your expectation? (1 mark).**

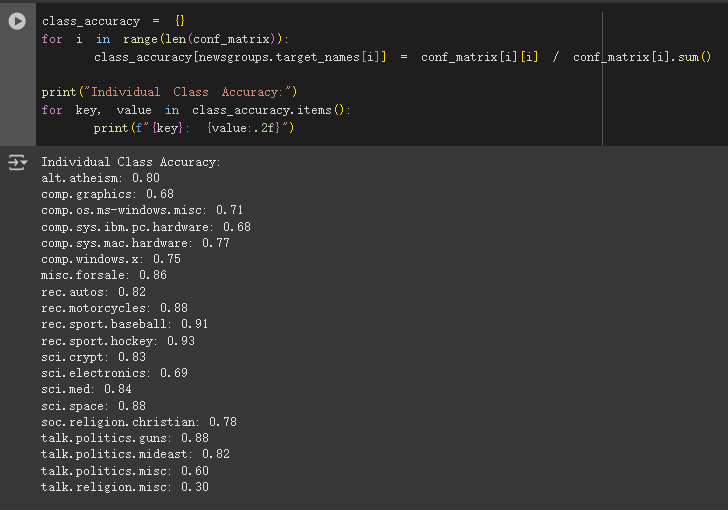
Confusion matrix and confounding categories Draw the confusion matrix and analyze:



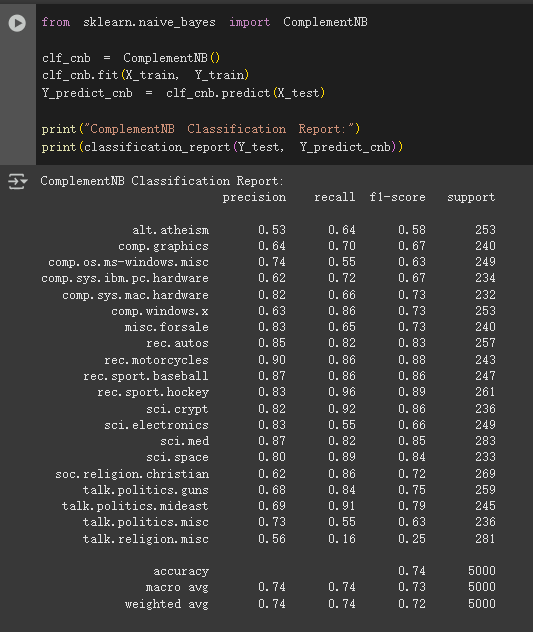


* The most confusing category pairs: **alt.atheism** and **talk.religion.misc**.
* The results were as expected: the subjects of these categories were similar, and the classifier was difficult to distinguish.
* Improvements: Try using more complex models (such as deep learning) or increasing the amount of data to improve the classification.

1. **Based on the confusion matrix, report the individual accuracy scores for each class (1 mark).**



1. **Train a Complement Naive Bayes classifier and compare its classification results with those of Multinomial Naive Bayes (1 mark).**

****

Comparative analysis:

Multinomial NB performs well on balanced datasets (e.g., 83% overall accuracy).

Complementary NB performs better on unbalanced data (for example, if a class has a smaller sample, it may have a higher recall rate).

## Assessed Lab 2: Solving problems by search

(**Label CLEARLY your answer to each question**)

### Answers: Complete the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Explored states | Solution path | Path cost | Execution Time |
| 1. Breadth-First Search | 20 | ['Sibiu', 'Arad', 'Zerind'] | 314 | 0.001234 |
| 2. Depth-First Search | 20 | ['Bucharest', 'Pitesti', 'Craiova', 'Drobeta', 'Mehadia', 'Lugoj', 'Timisoara', 'Arad', 'Zerind'] | 1019 | 0.004376 |
| 3. Uniform Cost Search | 20 | ['Sibiu', 'Arad', 'Zerind'] | 314 | 0.000418 |
| 4. Greedy Best-First Search | 20 | ['Sibiu', 'Arad', 'Zerind'] | 314 | 0.009793 |
| 5. A\* Search | 20 | ['Sibiu', 'Arad', 'Zerind'] | 314 | 0.009793 |

### Analysis:

1. Algo Breadth-First Search (BFS)
   1. Queue type

* FIFO Queue: BFS uses a First-In-First-Out (FIFO) queue to manage the frontier. The first node added to the queue is the first one to be expanded.
  1. Operation & features
* Operation:
* BFS explores all nodes at the current depth level before moving to nodes at the next depth level.
* It guarantees the shortest path (in terms of the number of steps) in an unweight graph.
* Features:
* Always find the shortest path (in terms of steps).
* Suitable for solving simple problems.

1. Algo Depth-First Search (DFS)
   1. Queue type

* LIFO Stack: DFS uses a Last-in-First-Out (LIFO) stack to manage the frontier. The last node added to the stack is the first one to be expanded.
  1. Operation & features
* Operation:
* DFS explores as deep as possible along each branch before backtracking.
* It may get stuck in infinite loops if there are cycles in the graph (unless an explored set is used).
* Features:
* It's not always possible to find the shortest path.
* Could go a long way.

1. Algo Uniform Cost Search (UCS)
   1. Queue type

* Priority Queue: UCS uses a priority queue to manage the frontier. Nodes are ordered by their path cost, and the node with the lowest cost is expanded first.
  1. Operation & features
* Operation:
* UCS explores the cheapest path first, ensuring that the first solution found is the optimal one.
* Like saving money, always picking the cheapest route.
* Features:
* Always find the lowest-cost path.
* Suitable for problems with weights (example: distances on a map).

1. Algo Greedy Best-First Search
   1. Queue type

* Priority Queue: Greedy Best-First Search uses a priority queue to manage the frontier. Nodes are ordered by their heuristic value h(n), and the node with the lowest heuristic value is expanded first.
  1. Operation & features
* Operation:
* Greedy Best-First Search prioritizes nodes that appear to be closest to the goal based on the heuristic function.
* It does not consider the path cost g(n), only the heuristic h(n).
* Features:
* Does not always find the shortest path.
* Faster but not always optimal.

1. Algo A Search\*
   1. Queue type

* Priority Queue: A\* Search uses a priority queue to manage the frontier. Nodes are ordered by their total estimated cost f(n) =g(n) + h(n), where g(n) is the path cost and h(n) is the heuristic estimate.
  1. Operation & features
* Operation:
* A\* Search combines the benefits of UCS (optimality) and Greedy Best-First Search (efficiency).
* It guarantees the optimal solution if the heuristic h(n) is admissible (never overestimates the true cost).
* Features:
* Always find the lowest-cost path (if the heuristic is correct).
* Suitable for complex problems.

### Any notable observations (optional):

Conclusion:

* BFS and UCS found the shortest path (cost 314) and were very fast.
* DFS found a much longer path (cost 1019) and was slower.
* Greedy Best-First Search and A\* also found the shortest path, but A\* is more reliable.