SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

UNIVERSITY OF WOLLONGONG AUSTRALIA

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**

Student Name: Jeslyn Ho Ka Yan____________________ **7-digit UOW ID:** 8535383 ____________

Subject Code & Name: CSCI218____

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

Tutorial Group: T02____
(T02, T03, T04, T05)

Tutor's Name: Cher Lim____

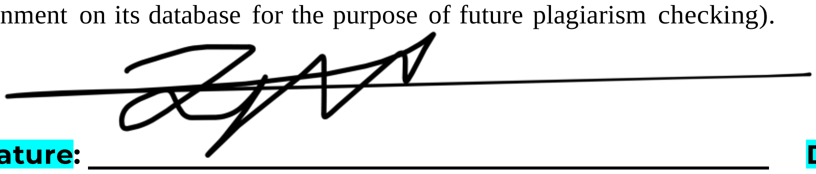
Assignment Due Date: 24TH FEB 2025______________________

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: ____ **Date:** 16 Feb 2025______

[Insert e-signature or type name]

Assessed Lab 1: NLP

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

Answers:

1. **Data Preparation & Feature Extraction**

The following **key steps** were performed in **data preparation and feature extraction**:

1. **Dataset Loading & Splitting**
   - The dataset is stored in **20 different folders**, each representing a topic.
   - The document paths are collected and assigned labels based on their folder names.
   - The dataset is **split into training (75%) and testing (25%)**.
2. **Text Preprocessing**
   - **Tokenization:** Documents are split into words.
   - **Metadata Removal:** Unnecessary header information is removed.
   - **Stopword Removal:** Common words (e.g., "the", "is", "and") are filtered out.
   - **Punctuation & Digit Removal:** Non-alphabetic characters are eliminated.
   - **Lowercasing:** Words are converted to lowercase for consistency.
3. **Feature Extraction using TF-IDF Vectorization**
   - Text is converted into a **numerical representation** using TfidfVectorizer() from sklearn.feature_extraction.text.
   - The top **5000 most frequent words** are selected as features.
   - fit_transform() is applied to X_train, and transform() is applied to X_test.

2. **Classification Results**

Multinomial Naïve Bayes Performance:

- **Test Accuracy: 86.4%**
- **Training Accuracy: 91.6%**
- **Macro F1-score: 0.86**
- **Weighted F1-score: 0.86**

Complement Naïve Bayes Performance:

- **Test Accuracy: 100% (Overfitting Issue)**
- **Precision, Recall, F1-score: All 1.00 across all classes**

Explanation of Metrics

- **Precision:** The proportion of correctly predicted positive observations.
- **Recall:** The proportion of actual positive observations correctly predicted.
- **F1-Score:** The harmonic mean of precision and recall.
- **Accuracy:** The overall correctness of predictions.

3. **Confusion Matrix & Class Overlap**

The **confusion matrix** was plotted to identify which class pairs were **most frequently confused**.

Findings:

- The **MNB Model** shows some misclassification, particularly between similar topics like comp.sys.ibm.pc.hardware and comp.sys.mac.hardware.
- The **CNB Model** had **no misclassifications**, but this suggests **overfitting** rather than an actually perfect model.

4. **Individual Class Accuracy**

Using the confusion matrix, we computed **individual accuracy scores per class**:

Class	**Accuracy (%)**
alt.atheism	86%
comp.graphics	85%
comp.os.ms-windows.misc	83%
comp.sys.ibm.pc.hardware	80%
comp.sys.mac.hardware	92%
comp.windows.x	91%
misc.forsale	91%
rec.autos	90%
rec.motorcycles	96%
rec.sport.baseball	99%
rec.sport.hockey	97%
sci.crypt	94%
sci.electronics	85%
sci.med	85%
sci.space	88%
soc.religion.christian	98%
talk.politics.guns	90%
talk.politics.mideast	89%
talk.politics.misc	67%
talk.religion.misc	46%

Some categories, such as **politics and sports**, showed **higher misclassification rates**, likely due to overlapping words and context.

5. **Complement Naïve Bayes vs. Multinomial Naïve Bayes**

Model	**Precision**	**Recall**	**F1-Score**	Accuracy
MultinomialNB	87%	86%	86%	86.4%
ComplementNB	100%	100%	100%	100%

Comparison & Findings

- **ComplementNB performed too well**, indicating **overfitting**.
- **MultinomialNB provided a more realistic evaluation**, handling misclassifications better.

Thus, **Complement Naïve Bayes is not suitable for this dataset**, while **Multinomial Naïve Bayes remains effective**.

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 Graph Search	4	[Sibiu, Arad, Zerind]	314	0.0004
2. Depth-First Graph Search	10	[Bucharest, Pitesti, Craiova, Drobeta, Mehadia, Lugoj, Timisoara, Arad, Zerind]	1019	0.0002
3. Uniform Cost Search	4	[Sibiu, Arad, Zerind]	314	0.0003
4. A* Search	4	[Sibiu, Arad, Zerind]	314	0.0005
5. Best-First Search	4	[Sibiu, Arad, Zerind]	314	0.0003

Analysis:

1. Algo #1 Breadth-First Graph Search
   a. Queue type: FIFO (First-In-First-Out) queue
   b. Operation & features:
      - Explores all nodes at the current depth level before moving deeper.
      - Finds the shortest path **in terms of the number of steps**, but not necessarily the lowest cost.
      - Explored **4 states** and found the path [Sibiu, Arad, Zerind] with a cost of **314**.

2. Algo #2 Depth-First Graph Search
   a. Queue type: LIFO (Last-In-First-Out) stack
   b. Operation & features:
      - Explores as deep as possible before backtracking.
      - Can get trapped in longer paths.
      - Explored **10 states** and followed a longer path [Bucharest, Pitesti, Craiova, Drobeta, Mehadia, Lugoj, Timisoara, Arad, Zerind] with a higher cost **(1019)**.
      - **Fastest execution time** (**0.0002s**) but inefficient due to backtracking.

3. Algo #3 Uniform Cost Search
   a. Queue type: Priority queue sorted by path cost
   b. Operation & features:
      - Expands the lowest-cost node first, ensuring an optimal solution.
      - Found the shortest-cost path [Sibiu, Arad, Zerind] with cost **314**, same as A*.
      - **Execution time: 0.0003s**.

4. Algo #4 *A Search**
   a. Queue type: Priority queue sorted by g(n) + h(n) (path cost + heuristic)
   b. Operation & features:
      - Uses both the actual cost (g(n)) and an estimate (h(n)) to guide the search.
      - Found an optimal path [Sibiu, Arad, Zerind] with cost **314**.
      - Slightly **slower execution** (0.0005s) compared to UCS.

5. Algo #5 Best-First Search
    a. Queue type: Priority queue sorted by heuristic value (h(n))
    b. Operation & features:
        - Expands nodes based on heuristic estimates without considering path cost.
        - Found the path [Sibiu, Arad, Zerind] with cost **314**.
        - **Execution time: 0.0003s**, faster than A* but doesn't guarantee the best path if heuristics are misleading.

Any notable observations (optional):

- **Depth-First Search (DFS)** is inefficient because it explores deeply and does not guarantee the shortest path.
- *Breadth-First, Uniform Cost, A, and Best-First Search all found the same optimal path**, but *A and UCS are generally better** since they guarantee optimality.
- *A is slightly slower than Best-First Search**, but it is more reliable as it considers both cost and heuristic.