Assignment-1

Anand Mishra

January 19, 2026

Note:

- Please read the problem statements carefully. Failure to follow the given instructions (e.g., naming conventions, requirments) may result in zero marks.
- Plagiarism is strictly prohibited. Any form of plagiarism, such as copying directly from LLMs or from another student, may result in zero marks for this assignment. Your submitted code will be compared against LLM-generated code as well as other student submissions.
- Write detailed and meaningful comments in your code. This is required to demonstrate your understanding of the implementation and to help assess originality. Submissions without proper comments will receive only 50% of the total marks.
- You should submit a single Zip File rollNumber_A1.zip. This zip file should contain the following files: RollNumber_prob1.py, RollNumber_prob1.log, RollNumber_prob1.txt, RollNumber_prob2.py, RollNumber_prob3.py, RollNumber_prob4.pdf. The sample blank files for a sample roll number B15CS001 have been shared in Google Classroom.
- DEADLINE: Feb 15, 2026 **(Firm Deadline, No Extension)**
-

PROBLEM 1: REGGY++

We have seen a toy chatbot Reggy in the class that is designed solely on regular expression and only understand pattern. Extend this chatbot (but still using regular expression) so that:

(i) It asks your birthday (in any possible formate including but not limited to mm-dd-yy,dd-mm-yy, dd-mm-yyyy, dd *Month Name in full or short* YYYY, and calulates your age. **(3 Points)**
(ii) It asks for your mood and based on mood responses appropriately. It deals with minor typing or spelling mistakes.**(3 Points)**
(iii) Add functionality to detect surname from full name. **(2 Points)**
(iv) Run your chatbot multiple times with multiple possible inputs and variation; and comment on its naturalness. **(2 Points)**
Deliverables:

1. **Python Source Code:** A single Python file named `RollNumber_prob1.py`.
   - The program must be executable from the terminal using:

     ```
     python RollNumber_prob1.py
     ```

   - No external dependencies are allowed beyond Python's standard library.
   - All chatbot logic must be implemented using regular expressions and basic string processing, i.e. your code should only use *import re* and *from datetime import date*.

2. **Execution Log File:** A log file named `RollNumber_prob1.log` containing transcripts of multiple chatbot runs.
   - Each run must clearly show:
     - User input
     - Chatbot response
   - Logs should demonstrate:
     - Different date formats
     - Different mood expressions (including typos)
     - At least one failure or ambiguous case
   - Each run should be separated clearly (e.g., by a divider or run number).

3. **Reflection on Naturalness:** A plain text file named `RollNumber_prob1.txt` containing the student's reflection.
   - Length: approximately 300–500 words.
   - Must discuss:
     - How natural or unnatural the chatbot feels
     - Strengths of regex-based interaction
     - Limitations and failure cases

(NOTE: not following instructions above may lead to 0 marks).

PROBLEM 2: TOKENIZATION

Write Byte Pair Encoding Tokenization from scratch. It should take a corpus (from an input text file) and number of merges K and return vocabulary. **(5 points)**

1. **Python Source Code** A single Python file named `RollNumber_prob2.py`
    - The program must be executable from the terminal as:

      ```
      python RollNumber_prob2.py corpus.txt
      ```

    - The file `corpus.txt` will contain the training corpus, with one sentence or one word sequence per line.
    - The code must implement Byte Pair Encoding (BPE) from scratch.
    - Only Python standard libraries are allowed (e.g., `collections`, `re`, `sys`).
    - Use of any external NLP or tokenization library is strictly prohibited.

PROBLEM 3: NAIVE BAYES FOR SENTIMENT CLASSIFICATION

You are given two text files:

- `pos.txt`: contains positive sentiment sentences, one sentence per line.
- `neg.txt`: contains negative sentiment sentences, one sentence per line.

Your task is to develop a Naive Bayes classifier from scratch to perform sentiment classification.
Task Description

1. Read the positive and negative files and construct a Naive Bayes model.
2. Use any reasonable split of the data for training and validation.
3. Tokenize sentences using simple whitespace tokenization and lowercase all words.
4. Apply Laplace smoothing while estimating probabilities.
5. After training, the program should:
    - Ask the user to enter a sentence from the terminal.
    - Predict its sentiment.
    - Output either `POSITIVE` or `NEGATIVE`.

Constraints

- Implement the classifier from scratch.
- Only Python standard libraries are allowed.

- Use of any machine learning or NLP library is strictly prohibited.

Deliverable

- A single Python file named `RollNumber_prob3.py`
- The code must run as:

```
python RollNumber_prob3.py
```

- The program should train the model and then enter an interactive mode for sentiment prediction.

PROBLEM 4: SPORTS OR POLITICS

: You have to design a classifier that reads a text document and classify it as Sport or Politics using any of your favorite machine learning techniques. For feature representation you can use: n-grams, TF-IDF, Bag of Words. You should compare atleast three ML Techniques for this task. Write a detailed report (minimum 5 page) starting from how did you collected data for this task, your dataset description and anlysis, techniques in brief and their quantative comparisions, limitations of your system. You should submit a GitHub page with all details for this problem.
Deliverable: Original Report and a GitHub Page.