

Assignment 3

Due Date: 23:59 July 31, 2025

Question 1 (25%)

In our discussion of sentiment analysis (SA), we installed and applied the Flair SA software tool to determine the sentiment of short English messages. This question asks you to explore alternative SA tools to address a similar problem. Specifically, you are to implement a Jupyter Notebook that installs and imports a different SA tool (such as BERT, FinBERT, VADER, etc.). Your task is to input ten English sentences, compute their sentiment scores using the chosen tool, and then output the average sentiment score of these ten sentences.

Question 2 (25%)

In Lecture Notes LN5, we implemented a Jupyter Notebook to determine the sentiment of movie reviews in the `nlk.corpus` by training a `NaiveBayesClassifier`. This question asks you to modify the Notebook as follows:

Instead of using the `list words_features` directly to construct the inputs for the classifier, perform the following preprocessing steps on the list of words:

1. Remove all punctuation marks.
2. Apply stemming to reduce words to their root forms.
3. Remove all stop-words.

After completing these preprocessing steps, continue to construct the feature sets, train the classifier, and evaluate its accuracy on the test set, similar to the procedure outlined in the lecture notes.

Question 3 (25%)

We have previously implemented a Q-learning algorithm to play the Cart-Pole game from Gym. This question requires you to implement a Monte Carlo algorithm to play the same game.

The core of your algorithm involves four weights: w_0 , w_1 , w_2 , and w_3 , corresponding to the four features of the environment's observations:

- Cart Position
- Cart Velocity
- Pole Angle
- Pole Angular Velocity

At each step, compute the weighted sum: and decide the action based on the sign of this sum (e.g., move left if negative, move right if positive).

Since the optimal weights are unknown, initialize each weight with a random value and use these weights to play the game. Record the total number of steps the weights can sustain. To find better weights, repeat this process 10,000 times, each time recording the weights that achieve the highest performance.

Finally, your program should output the best set of weights found and the total number of steps achieved with these weights.

Question 4 (25%)

In our implementation of a movie recommendation system, we initially used Pandas' groupby method to ensure that for each user, 90% of their ratings are included in `X_train`, and the remaining 10% in `X_test`.

Alternatively, we can achieve this through a randomization procedure. Specifically, we can use an appropriate method from Python's random library to set a new column 'for_testing' in the ratings DataFrame such that, for each row, 'for_testing' is assigned True with probability 0.1 and False with probability 0.9.

This way, for each user, the expected proportion of his/her ratings with 'for_testing' equal to True is approximately 10%.

Your task is to modify the Jupyter Notebook provided in the lecture notes LN7 to implement this randomized procedure, i.e., set the 'for_testing' column to the ratings DataFrame using the described method. Then, run the revised notebook to evaluate and print the results for `X_test` and `X_train` using the function `evaluate_rating()`.

Submission instructions

Submit 4 Jupyter Notebook files to the corresponding Assignment 3 submission links based on the Question number. Please note that the .ipynb file should already be run on Google Colab with the outputs. Please add your text answer in the text cell. All the submissions via Moodle.