MTU
Ollscoil Teicneolaíochta na Mumhan
Munster Technological University

MACHINE LEARNING

Assignment 1: Bayesian Classification

DUE DATE

This assignment should be submitted to Canvas before 11:59pm on **Friday 21/10/2022**.

Please submit a single ZIP file with your student number and name in the filename. Your submission should contain **exactly 2 files**:

- A detailed documentation of all code you developed, including the tests and evaluations you carried out. Please make sure that you include a .pdf document with every result you produce referencing the exact subtask and lines of code it refers to.
- All Python code you developed in a single .py file that can be executed and that generates the outputs you are referring to in your evaluation. The file needs to be readable in a plain text editor, please do NOT submit a notebook file or link. Please also make sure that you clearly indicate in your comments the exact subtask every piece of code is referring to.

Please do NOT include the input files in your submission.

You can achieve a total of 30 points as indicated in the tasks.

OBJECTIVE

The Excel file “movie_reviews.xlsx” on Canvas contains movie reviews from IMDb along with their associated binary sentiment polarity labels (“positive” or ”negative”) and a split indicator to determine if you are supposed to use the corresponding review for training your classifier (“train”) or for the evaluation (“test”).

The goal of this assignment is to create and evaluate a Naïve Bayesian classifier that can read a movie review and decide, if the review author would rate the movie as positive or negative based on the text entered.

TASK 1 (splitting and counting the reviews, 8 points)

Create a function that reads the file and separates it into training data and evaluation data based on the split indicator. The function should return four lists:

- Training data, containing all reviews of the training set [1 point]
- Training labels, containing all associated sentiment labels for the training data [1 point]
- Test data, containing all reviews of the test set [1 point]
- Test labels, containing all associated sentiment labels for the test data [1 point]

Further to that, the function should print on the console

- the number of positive reviews in the training set [1 point]
- the number of negative reviews in the training set [1 point]
- the number of positive reviews in the evaluation set [1 point]
- the number of negative reviews in the evaluation set [1 point]

TASK 2 (extract relevant features, 5 points)

Create a function that goes through all reviews in the training data extracted in task 1. Some of the reviews contain non-alphanumeric (e.g. ".", ",", "!", etc.) characters that should be removed before processing. Remove all such extra characters from the reviews [1 point], convert the reviews to lower case [1 point] and split the review content into individual words [1 point].

Now count the number of occurrences of each word in the training set [1 point]. The function should take the data set (i.e. the training data constructed in task 1) as input parameter. Further to that, it should have an input parameter for specifying the minimum word length and the minimum number of word occurrence. Using this mapping of words to number of occurrences in the training set, extract all the words from the reviews that meet these minimum requirements [1 point]. The function should return these words as list.

TASK 3 (count feature frequencies, 2 points)

Use the function created in task 2 to extract the set of all words of a minimum length and with a minimum number of occurrences from the reviews in the training set. Now create a function that goes through all positive reviews in the training set and counts for each of these words the number of reviews the word appears in [1 point]. Do the same for all negative reviews as well [1 point].

The function should take the review set to be searched and the set of words to look for as input parameters and should return as output a dictionary that maps each word of the input set to the number of reviews the word occurred in in the input set. If a word is not found in any review in the input set it should map to 0.

TASK 4 (calculate feature likelihoods and priors, 2 points)

Consider each word extracted in task 2 as a binary feature of a review indicating that a word is either present in the review or absent in the review. Using the function created in task 3 to count the number of reviews each of these features is present in calculate the likelihoods

$$P[word\ is\ present\ in\ review | review\ is\ positive]$$

and

$$P[word\ is\ present\ in\ review | review\ is\ negative]$$

for each word in the feature vector.

Create a function that calculates these likelihoods for all words applying Laplace smoothing with a smoothing factor $\alpha = 1$ [1 point]. The function should take the two mappings created in task 3 and the total number of positive/negative reviews obtained in task 1 as input and return a dictionary mapping each feature word to the likelihood probability that a word is present in a review given its class being either positive or negative.

Also calculate the priors

$$P[review\ is\ positive]$$

and

$$P[review\ is\ negative]$$

by considering the fraction of positive/negative reviews in the training set [1 point].

TASK 5 (maximum likelihood classification, 2 points)

Use the likelihood functions and priors created in task 4 to now create a Naïve Bayes classifier for predicting the sentiment label for a new review text [2 points]. Remember to use logarithms of the probabilities for numerical stability.

The function should take as input the new review text as string as well as the priors and likelihoods calculated in task 4. It should produce as output the predicted sentiment label for the new review (i.e. either “positive” or “negative”).

TASK 6 (evaluation of results, 11 points)

Create a k-fold cross-validation procedure for splitting the training set into k folds and train the classifier created in tasks 2-5 on the training subset [1 point]. Evaluate the classification accuracy, i.e. the fraction of correctly classifier samples, on the evaluation subset [1 point] and use this procedure to calculate the mean accuracy score [1 point].

Compare different accuracy scores for different choices (1,2,3,4,5,6,7,8,9,10) of the word length parameter as defined in task 2 [1 point]. Select the optimal word length parameter [1 point] and evaluate the resulting classifier on the test set extracted in task 1.

The final evaluation should contain:

- The confusion matrix for the classification [1 point]
- The percentage of true positive [1 point], true negatives [1 point], false positives [1 point] and false negatives [1 point]
- The classification accuracy score, i.e. the fraction of correctly classified samples [1 point]

Task 7 (optional, no marks)

Choose a movie you like and a movie you hate and write a review for both. Try the classifier on your review and see if the predicted sentiment score matches your own sentiment.