CS589 Machine Learning - Fall 2020

Homework 2: Naïve Bayes and Logistic Regression

Due: September 25th, 11:59 PM

Getting Started: You should complete the assignment using your own installation of Python 3.6. Download the assignment archive from Moodle and unzip the file. This will create the directory structure as shown below. You will write your code under the Submission/Code directory. Make sure to put the deliverables (explained below) into the respective directories.

```
HW04
--- Data
    |-- Women's Clothing E-Commerce Reviews Dataset
    Code
    |-- NB_Template.py
    |-- LR_Template.py
--- Submission
    |--Code
    |--Figures
    |--Predictions
```

If you are stuck on a question consider attending the office hours of the TAs.

Data Sets: For this homework, we'll use Women's Clothing E-Commerce Reviews Dataset. This is a real dataset revolving around the reviews written by customers. There are following features given to you for this dataset and their explanation:

- Age: Positive Integer variable of the reviewers age.
- Title: String variable for the title of the review.
- Review Text: String variable for the review body.
- Division Name: Categorical name of the product high level division.
- Department Name: Categorical name of the product department name.
- Class Name: Categorical name of the product class name.

Note that in this assignment we will **Only** use **Review Text** in your models for all the questions, however in last question you can use any of the given features. The labels are either positive or negative which says if the review is positive or negative. This task is called sentiment analysis.

Dataset	Training Cases	Validation Cases	Test Cases	Target
Women's Clothing E-Commerce Reviews	13193	3299	4123	Positive or Negative

Deliverables: This assignment has two types of deliverables: a report and code files.

- **Report:** The solution report will give your answers to the homework questions (listed below). The maximum length of the report is 5 pages in 11 point font, including all figures and tables. You can use any software to create your report, but your report must be submitted in PDF format.
- **Code:** The second deliverable is the code that you wrote to answer the questions, which will involve training classifiers and making predictions on held-out test data. Your code must be Python 3.6 (no iPython notebooks, other formats or code from other versions). You may create any additional source files to perform data analysis. However, you should aim to write your code so that it is possible to re-produce all of your experimental results exactly by running *python run_me.py* file from the Submissions/Code directory. Remember to comment your code. Points will be deducted from your assignment grade if your code is difficult to reproduce!

Submitting Solutions: When you complete the assignment, place your final code in Submission/Code. If you used Python to generate plots then place them in Submission/Figures. Finally, create a zip file called 'Submission.zip' from your 'Submission' directory only (do not include 'Data' directory). Only .zip files will be accepted for grading code (not .tar, .rar, .gz, etc.). You will upload your .zip file of code and your pdf report to Gradescope for grading. Further instructions for using Gradescope will be provided on Piazza and discussed in class.

Academic Honesty Statement: Copying solutions from external sources (books, web pages, etc.) or other students is considered cheating. Sharing your solutions with other students is considered cheating. Posting your code to public repositories like GitHub is also considered cheating. Any detected cheating will result in a grade of 0 on the assignment for all students involved, and potentially a grade of F in the course.

Note: You can only use sklearn's roc_auc_score and CountVectorizer methods

1 Naïve Bayes and Bag of Words [30 points]

In this section you will develop a Naïve Bayes Classifier for a sentiment analysis task. The dataset you will use is the Women's Clothing E-Commerce Reviews (WCECR) Dataset. At each entry/row of the input data X, only the **Review Text** feature is going to be used for classifying whether the review is positive or negative.

We will use Area Under the Receiver Operating Characteristic Curve (ROC AUC) for evaluating our model here, you can use sklearn's roc_auc_score method for calculating this, for more information on roc auc you can look at this link. And we will implement the function for confusion matrix in this question.

1. [10 points] You will be using Naïve Bayes to classify the *review text* into sentiment labels. First, a Bag of Words (BoW) model is needed to generate a numeric sequence $\mathbb{X} = (X_1, X_2, ..., X_n)$ for each entry of the review text, X_i is the count of the word at position i of the BoW model vocabulary in the review text. For example, if the BoW model vocabulary is $\mathbb{V} = \{'good', 'bad', 'test', 'this', 'perfect', 'is'\}$, the review text of *'this is good'* will be represented by a sequence $\mathbb{X} = (1, 0, 0, 1, 0, 1)$ in the BoW model.

   For this question, you need to implement the Bag of Words model from scratch, and use the implemented model to build a vocabulary $\mathbb{V} = \{v_1, v_2, ..., v_{10}\}$ of size 10 using the first 100 entries in **Data/X_train.csv**, then transform the next 100 entries into a 100 by 10 representation matrix $\mathbb{X} = \{X_1, X_2, ..., X_{100}\}$. For this question, you need to report the total word count of the vocabulary $\mathbb{V}$ in the 100 by 10 representation matrix, and the word count of the $\mathbb{V}$ in the representation matrix $\mathbb{X}$ should be a vector of size 10. (hint: sum the representation matrix along the first axis)

   Note:

   (a) In order to have consistent results for grading, you need to sort the vocabulary **alphabetically**, and build the vocabulary based on the word frequency.

(b) When building the vocabulary or transforming texts into representation matrix, remember to convert the text to lower case, remove the punctuation, empty spaces and 1-letter words from texts.

2. [5 points] Let the vocabulary size be D, then the input data $\mathbb{X} = (X_1, X_2, ..., X_D)$ for the Naïve Bayes model would be of shape (N, D), where N is the number of data entries. The sentiment labels $Y = \{0, 1\}$ where 0 means negative review and 1 means positive review. Derive $P(Y = 1|\mathbb{X})$ using Multinomial Naïve Bayes (assuming each word in the review is independent given the review sentiment), and remember to adjust the formula to accommodate the possibility of some words having a zero count for a class.

3. [10 points] For this question, you may use **CountVectorizer** from sklearn to build the Bag of Words model and implement the Multinomial Naïve Bayes model using your answer from Q1.2. Train your model on the training dataset (**Data/X_train.csv** and **Data/Y_train.csv**), predict on the validation dataset (**Data/X_val.csv** and **Data/Y_val.csv**), and report the ROC_AUC score, f1_score, accuracy and confusion matrix of the validation dataset predictions. For evaluation metrics, you may use sklearn.metrics for all but the confusion matrix.

4. [5 points] Tune the hyperparameter β of the Naïve Bayes model and plot the ROC_AUC score against the hyperparameter β, report the plot and your analysis of the plot.

2 Probabilistic Classification [20 points + 5 Extra Credit]

You are provided a dataset $\mathcal{D} = \{(X, Y)\}$ with n samples, where $X \in \mathbb{R}^D$, with even D, and the output Y is binary. The features are real valued, and you know that they have strong pairwise correlations, that is X_i is highly correlated with X_{i+1}, for all odd i with $0 \leq i \leq D$. That is, X_1 is correlated with X_2, X_3 is correlated with X_4 and so on. Each pair of features of the form (X_i, X_{i+1}) with odd i is conditionally independent of all other features given the output Y. You may assume real valued features are normally-distributed.

1. [5 points] What is the form of the Bayes optimal classifier for this problem? Please write out the factorized form of it, using the domain information you are provided.

2. [5 points] What parameters would you need to estimate to construct the Bayes optimal classifier, assuming normal distributions for the features? How many are there?

3. [5 points] Write out the MLE estimates for these parameters as a function of the data. [Note: you may use sample estimates.]

4. [5 points] What are the risks, when data is sparse? What can be done to mitigate these risks? [Please keep your answer brief, 3 sentences at most.]

5. [Extra Credit: 5 points] Select a suitable prior for the class-conditional feature distributions and write out the MAP estimates of the parameters.

3 Logistic Regression Revisited [40 points]

In this section you will implement ℓ_2-regularized Logistic Regression classifier from scratch (you are not allowed to use any machine learning library here). Please use the provided code template `code/LR_template.py` which contains the implementation details.

The logistic regression L_2 regularization formula is defined as:

$$\frac{1}{n}\sum_{n=1}^{N} -(y^{(i)}\log(\hat{y}^{(i)}) + (1 - y^{(i)})\ log(1 - \hat{y}^{(i)})) + \lambda||w||_2^2$$

where:

$$\hat{y}^{(i)} = \sigma(w^T x^{(i)})$$

and:

$$Sigmoid\ function : \sigma(z) = \frac{1}{1+e^{-z}}$$

Note that you will use bag of word representation for the text given in column "Review Text" like in the previous question. For optimization, use gradient descent with constant learning rate. For your experiments, use the dataset given to you. The task is to learn a classifier that can separate positive vs negative reviews (the reviews given in "Review Text" column).
Note: Try to vectorize your code so it runs faster. ROC AUC (see Q1) will be used to evaluate your model here.

1. [20 points] Use the training set to train the classifier, and the validation set to optimize the following hyperparameters in the provided range.

   - regularization weight: Try values in the range from 0 *to* 0.2
   - number of iterations: Try values in the range from 10 *to* 1500. Note that you do not have to train each model from beginning.
   - learning rate: Try values in the range from 10^{-4} *to* 10

   Report the set of candidate values that you use for each model parameter during model selection. Make sure you try at least 4 different values for each model parameter during model selection. Think about how to select (optimize) these model hyper parameters using the validation set. Explain and implement your model selection procedure. Report the train-set, validation-set AUC for the classifier that you obtain using the final set of model hyper parameters.

2. [10 points] Set all model hyper parameters to the ones given by model selection. By fixing all other hyper parameters and changing one hyper parameter at a time, plot the training and validation AUC with different regularization weights, number of iterations and learning rates. For example, one plot should be learning rates (X-axis) vs. AUC (Y-axis), where all parameters other than learning rate will be fixed to the values that are chosen by model-selection in (1). Each plot will contain two curves (train-set AUC and validation-set AUC) please indicate these in your plot legends. You will report 3 such plots in total (for some plots like learning rate you might decide to make your x-axis log scale). Describe your findings, and discuss them. Discuss how some of the model hyper parameters are inter-related.

3. [10 points] Calculate and report the confusion matrix you get on the validation data with the best hyper parameters you found in step (1).

4 Train Your Best Model [10 points]

1. [10 points] Now, for the Women's Clothing E-Commerce Reviews Dataset, you may use any model you like that has been taught in class or covered in the homework. You are also free to use any feature in addition to "Review Text" column here. Build your model, train the model on the full training dataset, then generate predictions of the test set, and save the outputs into the **Submission/Predictions** folder. Finally, describe your model selection, why you selected the model, how you tuned/choosed the hyperparameters (if applicable), and report your best validation result in the form of a table like:
   You can use any library you want including sklearn in this question.

Metric	Precision	Recall	F1 Score	AUC
Validation Set				