Northeastern University, Khoury College of Computer Science

CS 6220 Data Mining — Assignment 6

Due: March 28, 2023(100 points)

YOUR NAME
YOUR GIT USERNAME
YOUR E-MAIL

Mining Reviews in Small-ish Datasets

The question in this section is best started with the starter kit Colab. You can copy and past the cells to a `*.py` Python3 file or or simply use the Colab in your own Google Drive.

We'll process datasets that are small enough that they can fit into CPU memory. We'll review the Amazon purchasing reviews dataset. Amazon has collaborated with Universities to aggregate an extensive set of buyer reviews data. Most of the datasets (including the subcategory datasets) cannot fit into CPU RAM, and an even smaller subset of machine learning algorithms can actually process them.

With this dataset, you will need to do the following for question number 2. Typically we would split the dataset into training / validation / testing dataset, but we will only split into training / testing dataset in this assignment.

1. With the Amazon magazines dataset, do the following. (Feel free to use the `scikit-learn` library).

   - Plot a histogram of the data with the number of positive and negative reviews.
   - Balance the data so that you are training with equal probabilities, $P(\text{Liked}) = P(\text{Not Liked}) = 0.5$. We will use the original distribution for the evaluation dataset.
   - Use machine learning models to predict whether or not a review as written will result in a "Liked" rating. Try changing different parameters, including the maximum vocabulary size. Also try normalization.
   - Try out Naïve Bayes, Decision Trees, Random Forests, and Logistic Regression machine learning algorithms

- Print out your best accuracy, precision, and recall numbers for each algorithm.

How do they compare? Which algorithms are overfitting?

Tips and Tricks

You can preprocess the data using files in `cs6220hw5.py`, which you can download here. This will remove *stop words*, punctuation, and ensure that the texts are all lowercase. Caution that this implementation is exceedingly slow and un-optimized. (Extra credit to anyone who implements a faster version.)

Featurize your inputs to the algorithm with `CountVectorizer()`. Make sure to set the maximum vocabulary size / number of features, `max_features`.

Evaluating Your Classifier Performance

In lecture, you will recall that you can calculate the ROC curve by computing the true positive rate (TPR = Probability of Detection, i.e. P_D) and false positive rate (FPR = Probability of False Alarm, i.e P_{FA}) at every threshold. In this homework, our function will take in predicted classifier scores (`scores` from Livermore Laboratory sensors) and the true labels (`labels` of whether or not laser tubes are calibrated).

ROC/AUC from min-FPR to max-FPR

We will implement a special function that efficiently plots the ROC curve and calculates the AUC *for a specified range* of FPR. Where your function differs from traditional ROC curve calculations is that it will compute only the true positive rate (i.e., TPR, or P_D) from the range starting from some minimum FPR (i.e., min-FPR, or min-P_{FA}, defaulted to 0) to some specified maximum FPR (i.e., max-FPR, or max-P_{FA}, required parameter).

In particular, it will then return all the FPR values and TPR values that can create a ROC curve in the specified FPR range, as well as the area under it (AUC). For example, I can specify max-FPR = 0.4, and our function will give us all the ROC scores in the range FPR = $[0, 0.4]$, inclusive of the end values. While there may be a way to leverage existing libraries (e.g., Scikit-Learn), please refrain from doing so (though you can feel free to check your answers with them.)

The function signature looks like the following:

```
def roc_range(scores, labels, maxfpr, minfpr = 0)
  '''
  Inputs:
    scores - predictions from any given classifier
    labels - the true label (either 0, 1) of the data
    maxfpr - the maximum false positive rate (FPR)
    minfpr (optional, default = 0) - the minimum false positive rate (FPR)
```

```
    Outputs:
      fpr_in_range - list of FPR values from minpfa to maxpfa
      tpr_in_range - list of true positive rate (TPR) values from minfpr to maxfpr
      auc_in_range - single value of area under curve from minpfa to maxpfa
    '''

    return fpr_in_range, tpr_in_range, auc_in_range
```

Try it Out On Some Data

We will check your answers against our data, which you can find here at the course website. To read this data, feel free to use this code:

```
import numpy as np
data = np.load("assignment6.npz")

# The data that you will read in
scores_small = data['scores_small']
scores_large = data['scores_large']
labels_small = data['labels_small']
labels_large = data['labels_large']
```

This unpacks the data, which has some small test data that you can use to try out your algorithm before running the analysis on the larger set of data. If interested, this data has been drafted from National Ignition Facility readouts. Please make your code readable for any data with the above generic signature.

Homework Questions

2. Plot the ROC curve and calculate the AUC for the following ranges:
   a) $P_{FA} \in [0, 1.0]$, the full range of thresholds
   b) $P_{FA} \in [0, 0.4]$
   c) $P_{FA} \in [0, 0.75]$
   d) $P_{FA} \in [0.25, 0.75]$
3. Your implementation notes:
   a) Describe your implementation. How would you sweep your thresholds? For each threshold, how would you calculate the PFA and PD? What is the runtime in big-$\mathcal{O}$ notation?
   b) Determine the runtime of your implementation in big-$\mathcal{O}$.
   c) Can you make your implementation run in $\mathcal{O}(N \log N)$?
4. What thresholds provide a *precision* of 0.9?

5. At this threshold, what is the *accuracy* of the classifier?

Submission Instructions

Submit your work to Gradescope. There, you will need to upload a PDF file with the written answers (including the plots) to all the questions above. As well, there will be a link to upload your Python code. Make sure that the signature matches the above signature as we will check it against other types of data.