DS-UA 202, Responsible Data Science, Spring 2024

Homework 2: Data Science Lifecycle & Transparency and Interpretability
Due at 11:59 pm on Friday, April 5, 2024

Objectives

This assignment consists of written problems and programming exercises on two modules: the data science lifecycle (M2) and transparency and interpretability (M3). The programming part of this assignment focuses on explaining classifiers and rankers, and on doing feature selection based on a set of explanations. You will use two open-source libraries, SHAP and ShaRP, to complete the programming portion of this assignment. We encourage you to carefully read the papers describing both approaches (included in the reader). Further, we worked with the SHAP and ShaRP in labs 7 and 8, respectively.

After completing this assignment, you will:

1. Understand how to use SHAP to generate locally interpretable explanations of classification decisions on a text corpus.
2. Learn how to use local explanations for feature selection, ultimately improving classification accuracy on a text corpus.
3. Understand how to use ShaRP to generate local explanations for ranked outcomes, and to aggregate these explanations over sets of instances.

You must work on this assignment individually. If you have questions about this assignment, please post a private message to all instructors on Piazza.

Grading

The homework is worth 75 points, or 10% of the course grade. You also have an opportunity to earn 15 points of extra credit (2% of the course grade). Your grade for the programming portion (Problems 3 and 4) will be significantly impacted by the quality of your written report for that portion. In your report, you should explain your observations carefully.

You are allotted 2 (two) late days over the term, which you may use on a single homework, or on two homeworks, or not at all. If an assignment is submitted at most 24 hours late -- one day is used in full; if it's submitted between 24 and 48 hours late -- two days are used in full.

Submission instructions

Provide written answers to all problems in a single PDF file created using LaTeX. (If you are new to LaTeX, Overleaf is an easy way to get started.) Provide code in answer to Problem 3 and 4 in a Google Colaboratory notebook. Both the PDF and the notebook should be turned in as Homework 2 on Gradescope. Please clearly label each part of each question. Name the files in your submission *abc123*_hw2.pdf and *abc123*_hw2.ipynb (replace *abc123* with your UNI).

Problem 1 (15 points): AI Ethics: Global Perspectives

In this part of the assignment, you will watch a lecture from the AI Ethics: Global Perspectives course and write a memo (500 words maximum) reflecting on issues raised in the lecture. You can watch either:

- "Content Moderation in Social Media and AI" (watch the lecture)
- "AI Ethics and Hate Speech" (watch the lecture)
- "Algorithmic Transparency in the Public Sector" (watch the lecture)
- "Do Carebots Care? The Ethics of Social Robots in Care Settings" (watch the lecture)

If you have not already registered, please register for the course at https://aiethicscourse.org/contact.html, specify "student" as your position/title, "New York University" as your organization, and enter the course number, DS-UA 202, in the message box.

In your memo, you should discuss the following:

- Identify responsible AI concerns surfaced in the lecture.
- Identify the stakeholders. In particular, which organizations, populations, or groups could be impacted by the RAI concerns discussed in the lecture?
- How does transparency and interpretability, or a lack thereof, affect users or other stakeholders? Are there black boxes?
- What incentives does the vendor (e.g., the data owner, company, or platform) have to ensure transparency and interpretability? How do these incentives shape the vendor's behavior?

Discuss anything else related to transparency and interpretability raised in the lecture.

Problem 2 (10 points) : Data science lifecycle

Alex, a data scientist in the human resources department of a technology company *MegaSoft*, is running job applicants through an already trained model that ranks applicants for job openings at the company. The applicant dataset has four features: *sex*, *race*, *experience* (measured in years), and performance on a *skills test*.

Alex conducts data profiling on the applicant dataset and produces the table below. She finds that all demographic groups perform comparably on the skills test (results are not shown), but that experience differs across groups. Further, she knows that every applicant must take a skills test to apply for a job at *MegaSoft*, but that some applicants do not report their experience.

	EXPERIENCE (years)				
	male		female & non-binary		all
	white	other	white	other	
N	2001	1065	634	300	4000
min	1	1	1	1	1
mean	5.70	5.66	7.40	7.91	6.12
median	5	5	6	6	5
max	26	35	44	40	44
NULL	80	63	60	42	225

(a) (3 points) To prepare the data to run through the ranking model, Alex replaces missing values (NULL) in the experience feature with the **overall mean value** for that feature in the dataset. She expects the ranking model to prioritize (rank higher) those applicants who score higher on the skills test and have more years of experience. Looking at the data profiling table above, which applicant group(s) may be disadvantaged by Alex's imputation method and why?

(b) (3 points) Propose an alternative data imputation method that may improve the ranking of individuals from the group(s) you identified in **(a)**.

(c) (4 points) The data imputation method described in **(a)** can introduce technical bias. Explain how this type of technical bias relates to pre-existing and emergent bias in *MegaSoft*'s hiring example. *Be concise and concrete.*

Problem 3 (40 points): Explaining text classification with SHAP

For this programming portion of this assignment, we will use a subset of the text corpus from the 20 newsgroups dataset. This is the dataset used in the LIME paper to generate the Christianity/Atheism classifier, and to illustrate the concepts. However, rather than explaining predictions of a classifier with LIME, we will use this dataset to explain predictions with SHAP.

(a) (5 points) Use the provided Colab template notebook to import the 20 newsgroups dataset from sklearn.datasets, importing the same two-class subset as was used in the LIME paper: Atheism and Christianity. Use the provided code to fetch the data, split it into training and test sets, then fit a TF-IDF vectorizer to the data, and train a SGDClassifier classifier.

(b) (5 points) Generate a confusion matrix (hint: use sklearn.metrics.confusion_matrix) to evaluate the accuracy of the classifier. The confusion matrix should contain a count of correct Christian, correct Atheist, incorrect Christian, and incorrect Atheist predictions. Use SHAP's

explainer to generate visual explanations for any 5 documents in the test set. The documents you select should include some correctly classified and some misclassified documents.

(c) (15 points) Use SHAP's explainer to study mis-classified documents, and the features (words) that contributed to their misclassification, by taking the following steps:

- Report the **accuracy** of the classifier, as well as the **number of misclassified documents**.
- For a document **doc_i** let us denote by **conf_i** the difference between the probabilities of the two predicted classes for that document. Generate a chart that shows **conf_i** for all misclassified documents (which, for misclassified documents, represents the magnitude of the error). Use any chart type you find appropriate to give a good sense of the distribution of errors.
- Identify all words that contributed to the misclassification of documents. (Naturally, some words will be implicated for multiple documents.) For each word (call it **word_j**), compute (a) the number of documents it helped misclassify (call is **count_j**) and (b) the total weight of that word in all documents it helped misclassify (**weight_j**) (sum of absolute values of **weight_j** for each misclassified document). The reason to use absolute values is that SHAP assigns a positive or a negative sign to **weight_j** depending on the class to which **word_j** is contributing. Plot the distribution of **count_j** and **weight_j**, and **discuss** your observations in the report.

(d) (**15 points**) In this problem, you will propose a feature selection method that uses SHAP explanations. The aim of feature selection is to improve the accuracy of the classifier.

- **Implement** a strategy for feature selection, based on what you observed in Problem **3(c)**. Report the accuracy of the classifier after feature selection. Describe your feature selection strategy and the results in your report.
- Show at least one example of a document that was misclassified before feature selection and that is correctly classified after feature selection. In your report, discuss how the explanation for this example has changed.

Hint: As you are designing your feature selection strategy, consider removing some features (words) from the training set.

Problem 4 (10 + 15 points): Explaining rankings with ShaRP

When multiple people apply for a loan, banks need to select whom to fund. They do so by calculating how likely a person is to repay the loan, rank the people on this likelihood, and then select the most likely among them. Ranking functions may be specified by a vendor (e.g., a bank or a college admissions office) or they may be learned. In this part of the assignment, we will use three designed ranking functions on the ACSIncome dataset to rank the people in the dataset. Then we will see how feature importance changes for different ranking functions and different people and groups. To calculate feature importance, we will use ShaRP, an open-source library we worked with during lab 8.

(a) (10 points) Use the provided Colab template notebook to load 1,000 individuals from ACSIncome, preprocess them, and rank them using 3 scoring functions specified in the template. Select the individual ***i*** at position 100 in the ranking that uses the first scoring function, **score_function_SCHL**, and answer the following questions:

- (i) What is the rank of individual ***i*** in the rankings that are based on the other two scoring functions, **score_function_WKHP** and **score_function_AGEP**?
- (ii) Compute the importance of ***i*** according to rank QoI for each of the 3 rankings. Do all features contribute to rank QoI equally? Which features are most important for each function? How does feature importance vary across ranking functions and why?
- Repeat this process for another individual.

(b) (Extra Credit, 10 points) Suppose that the bank issues loans to the top-20% of the applicants, and that they want to understand which features contribute to the ranking of these individuals (in aggregate), both overall and broken down by race.

To help the bank with this analysis, rank the applicants based on **score_function_SCHL** and select the top 20% of the applicants. Then perform the following steps:

- (i) Split the top-20% into two groups (white and non-white) using the race feature. We recommend you have two new different dataframes, one for each group. Define ShaRP objects for both groups and calculate feature importance according to rank QoI for each group separately. (Because the sample size is small, there is no need to use approximation to calculate feature importance, and so you can change sample_size to None.)
- (ii) Use the function **make_boxplot_top20**, to visualize feature importance for white and non-white individuals and plot it on separate boxplots.
- (iii) Compare feature importance between the 0-10% and 10-20% *for each racial group*. Is feature importance different across these two strata? Which features are more / less important for each stratum?
- (iv) Compare feature importances *across racial groups*. Which features are more / less important for each group? Hypothesize about why feature importance may be different across groups.

(c) (Extra Credit, 5 points) In this question, you are asked to visually present an explanation of the relative ranking of a pair of individuals, using feature importance. That is, we are asking you to design a visualization that would explain why some item ***i*** is ranked higher than some other item ***j*.** Feel free to use any ShaRP functionality for this, and any plot type. We recommend that you visualize the comparison between items at positions 200 and 300 according to **score_function_SCHL**, but you should feel free to pick some other pair of items in addition to this pair to showcase the usefulness of your visualization.