ECON861 Advanced Data Analysis(Spring 2024) Midterm Exam

Due: 8th April 2024 (11:59pm)

Name: ____________________

UID: ____________________

---

1. (100 points) Train a model to predict the programmers' quality based on review text.

   - Download the exam dataset (review_data.zip) the file is approximately 53 KB in size.
   - In the folder training_data, there is a dataset (dataset.csv) containing reviews of about 850 programmers. "Stars" indicate the quality of the programmer (1: bad, 2: average, 3: good).
   - Train a Naive Bayesian model using the dataset of reviews to predict stars based on review text, as discussed in class.
   - Save the trained model as a pickle file.
   - Your boss will use the pickle file you generated to predict the quality of 10,000 programmers using a file containing their review texts. You have a very small sample of these texts (sample_new.csv). Write code to predict the quality of programmers in the boss's file, and provide a short instruction on how to use your code.
   - Compare the performance of the Naive Bayesian model with a Random Forest model or a Logistic Regression model. Which model would you choose, and why? Include this discussion in the code instruction for your boss and explain your choice.
   - Unless you have 100% accuracy, there are some reviews in the dataset that are misclassified. Identify several misclassified reviews from the training dataset and speculate why they are misclassified. Discuss your findings and explain the limitations of the model. Include this discussion in the code instruction for your boss.
   - Suppose after you finish the program, your boss changes his/her mind (which happens a lot) and say the most important thing is to predict whether the programmer is really bad (1) or not (2 or 3). Rewrite the code to predict the quality of the programmers in the boss file using the new criteria. Save your original work in a folder named "original_program", and save the new programs and the new pickle files in a folder named "new_criteria".
   - Bonus question: Suppose your boss changes his/her mind again (which again, happens a lot more than you can imagine) and say that instead of predicting the stars, your program need to suggest one programmer to be hired. How would you modify your program? It would be great if you can write the code, but it would be great if you can explain how you would modify your program and include the explanation in the code instruction for your boss.

   Further Instructions:

   - Instead of arguing which model is the best, the most direct way to compare the models is to train all the models and validate using KFold validation. You can use the kfold_template.py developed in class, but it is not required. If you use it, include it in your repository.
   - Since you only have access to a sample or your boss's dataset, it is not good to use the sample dataset as validation. In other words, having a 100% accuracy in predicting the sample datasets does not mean that your model is good. And performing poorly on the sample dataset does not mean that your model is bad (But it's not a good sign either). You should use KFold validation to validate your model.

- You should hand in your work via Github. Please set your Github repository to a private repository and invite me to be your collaborator. DO NOT commit the dataset into the repository. Use the .gitignore file to exclude the dataset from the repository.
- Do not print out your answers and hand in hardcopies. You fail this class immediately if you do that.
- Your Github repository should contain all the files you would like to submit. You can organize your files in any way that makes sense to you. However, here is a suggested structure for your repository:
  - A .gitignore file to specify which files should be ignored by git. For example, you should not put the dataset and the other temporary files in your repository.
  - Two folders, "original_program" and "new_criteria". Each of them contains:
    - A folder "private_files", which contains the code for training the models (native bayesian, random forest, logistic regression......etc.) and other required files (such as kfold_template.py). When I run the Python files in this folder, it should output pickle files to be used in the boss's program.
    - A folder "boss_files", which contains the code that predicts the quality of the programmers using (sample_new.csv). You SHOULD also include the pickle files in this folder. When I run the Python files in this folder, it should output a csv file containing the predicted stars of the programmers ("sample_new_with_stars.csv").
  - An instruction file ("README.md") which contains the instructions for your boss. You should include all the instructions for the "original_program" and the "new_criteria" program in this file. This includes how to train the models, how the boss can use the trained models to make predictions.....etc. You should also include a discussion of the limitations.
- Even though it is assumed that your boss will only use your code for prediction, it makes sense to include also the instructions of how you train the models. This is because your colleague may need it in the future in case they are asked to take over your project.
- README.md is in markdown format so you can utilize sections, bullet points, and other markdown features to make it more readable.
- It is more important to attempt all the questions than to answer one part of the question perfectly. If you are not sure about how to answer a question, you should still attempt it. You will get partial credits for that. Also, if you find it difficult to finish all your analysis in Python, you can use other tools to finish the task and explain how you did it in the report. For example, you may merge two columns into one column in Excel if you do not know how to do it in Python using pandas, or you may manually fix some problems in the dataset if you do not know how to do it in Python. Or if you do not know how to separate the boss' files and the training files, you may hand in all the files in one folder. Depending on the nature of the tasks, your score may be affected, but it is still way better than not doing it at all.