Lab 3: Classification Challenge

This assignment is due to Gradescope by the beginning of lab next week (2:45p on 2/17). You may work with a partner on this lab – if you do, submit only one solution as a "group" on Gradescope.

1 Introduction

The purpose of this lab is to practice implementing shallow ML pipelines in a common format: the classification challenge.

Organizations and researchers often post datasets as public "competitions" to see how well machine learning models can perform on associated classification and regression tasks. People participate in these challenges for a variety of reasons. In some subfields of machine learning, beating the state-of-the-art performance on a widely-used dataset can be worth an academic publication alone. Others use these challenges ways to practice ML programming, contribute to citizen science initiatives, or win prizes (this ongoing competition is offering $150,000 in prizes for the models that can most accurately detect starfish in videos of the Great Barrier Reef).

This lab will let you experience an ML competition in a low-stakes environment. You will be provided with a labeled training dataset. Your task will be to train a machine learning model using any of the shallow learning techniques we have covered in class. At any point before the lab is due, you may upload your current model to Gradescope, where it will be automatically applied to the test set. Its performance will then be posted on an anonymized leaderboard so you can see how your model stacks up against the rest of the class.

You are encouraged to upload your model to the leaderboard early and often. While you won't have access to the test set labels, submitting allows you to check whether changes to your model have improved or reduced test set performance. In real ML competitions, it is common for teams to submit a "naive" model at the beginning of the competition to set themselves a baseline for improvement.

Ultimately, you will only be graded on 1) whether your model beats a simple 1-Nearest Neighbor classifier, 2) whether your code demonstrates meaningful effort to improve model performance through data preprocessing, model selection, and hyperparameter optimization, and 3) your answers to open-ended questions.

However, there will be **extra credit** awarded based on the leaderboard! If your model is among the top $p \leq 10$ by F_1 score when the lab is due next week, you will receive $11 - p$ extra credit points. For example, if you are in 1st place, you will receive 10 extra credit points.

2 Provided Files

- `Lab3.pdf`: This file
- `Lab3.py`: Code scaffold
- `Lab3_X_train.csv`: Training examples
- `Lab3_y_train.csv`: Training labels
- `Lab3_X_test.csv`: Test examples (note that test labels are not provided)
- `Lab3_questions.txt`: Open-ended questions

3 ML Task Description

The `Lab3_X_train.csv` file contains 10 years worth of daily weather observations from locations across Australia, one row per day. The features should be self-explanatory from the column headers. The `Lab3_y_train.csv` file contains one correct binary label for each observation: a 1 if it rained on the following day or a 0 if it did not. Your goal will be to create a ML model that, when given a new weather observation from `Lab3_X_test.csv`, can predict whether it will rain on the day after the observation. In other words, can you use machine learning to predict if it will rain tomorrow based on the weather today?

4 Instructions

Implement the `fit_predict()` function in `Lab3.py` to have the following behavior:

1. *(Provided)* Load the training examples, training labels, and test examples into local variables `X_train`, `y_train`, and `X_test`, respectively.
2. Preprocess `X_train` and `X_test` to impute missing features, encode nominal features, and appropriately scale all features (see tips on next page).
3. Train a `LogisticRegression`, `KNearestNeighbors`, or `SVC` model to perform the rain prediction task.
4. Use the model to predict and **return** labels for all examples in `X_test` as a Pandas series or NumPy array. There should be as many predicted labels as there are rows in `X_test`.

Your `Lab3.py` file may import any needed modules from `sklearn`, `numpy`, `scipy`, `pandas`, `matplotlib`, or `seaborn`. Do not import any other modules as they will not be available in the leaderboard environment.

4.1 Preprocessing Tips

Preprocessing `X_train` and `X_test` (step 2 above) will require several operations. Here are some suggestions to get you started:

- This dataset has some missing data points that need to be handled. You should not remove any examples just because they include missing data. Instead you should replace the missing values with something more reasonable of the correct type (float, string, etc.).
- This dataset has several nominal features that will need to be converted into numeric features. You can use the Scikit-Learn `OrdinalEncoder` class for ordinal encoding or the Pandas `get_dummies` method for one-hot encoding. You may choose either, but be careful not to accidentally encode any features that are already numerical.
- Remember to standardize your data and to apply the same transformations to *both* the training and test examples!

4.2 Additional Hints

These suggestions are strongly encouraged:

1. **Read `Lab3_questions.txt` before you start.** You will need to perform certain analyses to answer these questions, which you should plan for at the outset.
2. Divide `fit_predict()` into helper functions for modularity and easier testing.
3. Read the Scikit-Learn documentation for the models you use to find out what hyperparameters are available and how to set them.

4.3 Leaderboard

You may submit your `Lab3.py` to the Gradescope leaderboard as many times as you like before the due date next week. Submit your model often to see how small changes affect test set performance. When you upload your `Lab3.py` file, it will ask you to supply a "leaderboard name." You may enter your real name or choose a pseudonym to stay anonymous.

5 Final Submission

Once you are satisfied with your model's performance, submit **both** your `Lab3.py` and your completed `Lab3_questions.txt` to Gradescope.

6 Extra Credit Opportunity

If you find a bug anywhere in this lab, please inform Prof. Apthorpe. The first student(s) to find any particular bug will be given a small amount of extra credit. This will help make the course better for students in future years.