MiniProject 1: Machine Learning 101

COMP 551 (001/002), Winter 2020, McGill University

Please read this entire document before beginning the assignment.

Preamble

- This mini-project is **due on February 11th at 11:59pm.** Late work will be automatically subject to a 20% penalty, and can be submitted up to 5 days after the deadline. No submissions will accepted after this 5 day period.
- This mini-project is to be completed in groups of three. There are three tasks outlined below which offer one possible division of labour, but how you split the work is up to you. All members of a group will receive the same grade. It is not expected that all team members will contribute equally to all components. However every team member should make integral contributions to the project.
- You will submit your assignment on MyCourses as a group. You must register your group on MyCourses and any group member can submit. See MyCourses for details.
- You are free to use libraries with general utilities, such as matplotlib, numpy and scipy for Python. **However, you should implement everything (e.g., the models and evaluation functions) yourself, which means you should not use pre-existing implementations of the algorithms or functions as found in SciKit learn, etc.**

Background

In this miniproject you will implement two classification techniques—logistic regression and naive Bayes—and compare these two algorithms on four distinct datasets. The goal is to gain experience implementing these algorithms from scratch and to get hands-on experience comparing performance of different models.

Task 1: Acquire, preprocess, and analyze the data

Your first task is to acquire the data, analyze it, and clean it (if necessary). We will use two fixed and two open datasets in this project, outlined below.

- **Dataset 1 (Ionosphere)**: this is a dataset where the goal is to predict whether a radar return from ionosphere is 'good' or 'bad'. This radar data was collected by a system in Goose Bay, Labrador. Get it from:
  `https://archive.ics.uci.edu/ml/datasets/ionosphere`
- **Dataset 2 (Adult Data Set):** also known as "Census Income" dataset, this is a dataset where the goal is to predict the whether income exceeds $50K/yr based on census data. Get it from:
  `https://archive.ics.uci.edu/ml/datasets/Adult`
- **Dataset 3 and 4**: pick two additional datasets for classification from the UCI datasets at:
  `https://archive.ics.uci.edu/ml/datasets.php`

The essential subtasks for this part of the project are:

1. Download the datasets (noting the correct subsets to use, as discussed above).
2. Load the datasets into NumPy objects (i.e., arrays or matrices) in Python. Remember to convert the wine dataset to a binary task, as discussed above.
3. Clean the data. Are there any missing or malformed features? Are there are other data oddities that need to be dealt with? **You should remove any examples with missing or malformed features and note this in your report.** *For categorical variables you can use one-hot encoding.*
4. Compute basic statistics on the data to understand it better. E.g., what are the distributions of the positive vs. negative classes, what are the distributions of some of the numerical features? what are the correlations between the features? how does the scatter plots of pair-wise features look-like for some subset of features?

Task 2: Implement the models

You are free to implement these models as you see fit, but you should follow the equations that are presented in the lecture slides, and you must implement the models from scratch (i.e., you cannot use SciKit Learn or any other pre-existing implementations of these methods).

In particular, your two main tasks in the part are to:

1. Implement logistic regression, and use (full batch) gradient descent for optimization.
2. Implement naive Bayes, using the appropriate type of likelihood for features.

You are free to implement these models in any way you want, but you must use Python and you must implement the models from scratch (i.e., you cannot use SciKit Learn or similar libraries). Using the numpy package, however, is allowed and encouraged. Regarding the implementation, we recommend the following approach (but again, you are free to do what you want):

- Implement both models as Python classes. You should use the constructor for the class to initialize the model parameters as attributes, as well as to define other important properties of the model.
- Each of your models classes should have (at least) two functions:
  - Define a `fit` function, which takes the training data (i.e., $\mathbf{X}$ and $\mathbf{y}$)—as well as other hyperparameters (e.g., the learning rate and/or number of gradient descent iterations)—as input. This function should train your model by modifying the model parameters.
  - Define a `predict` function, which takes a set of input points (i.e., $\mathbf{X}$) as input and outputs predictions (i.e., $\hat{\mathbf{y}}$) for these points. *Note that you need to convert probabilities to binary 0-1 predictions by thresholding the output at 0.5!*
- In addition to the model classes, you should also define a functions `evaluate_acc` to evaluate the model accuracy. This function should take the true labels (i.e., $\mathbf{y}$), and target labels (i.e., $\hat{\mathbf{y}}$) as input, and it should output the accuracy score.
- Lastly, you should implement a script to run k-fold cross validation.

Task 3: Run experiments

The goal of this project is to have you explore linear classification and compare different features and models. *Use 5-fold cross validation to estimate performance in all of the experiments. Evaluate the performance using accuracy.* You are welcome to perform any experiments and analyses you see fit (e.g., to compare different features), **but at a minimum you must complete the following experiments in the order stated below**:

1. Compare the accuracy of naive Bayes and logistic regression on the four datasets.

2. Test different learning rates for gradient descent applied to logistic regression. Use a threshold for change in the value of the cost function as termination criteria, and plot the accuracy on train/validation set as a function of iterations of gradient descent.
3. Compare the accuracy of the two models as a function of the size of dataset (by controlling the training size).

   As an example, see Figure 1 here [1].

Note: The above experiments are the minimum requirements that you must complete; however, this project is open-ended. For example, you might investigate different stopping criteria for the gradient descent in logistic regression, develop an automated approach to select a good subset of features. You do not need to do all of these things, but you should demonstrate creativity, rigour, and an understanding of the course material in how you run your chosen experiments and how you report on them in your write-up.

Deliverables

You must submit two separate files to MyCourses (**using the exact filenames and file types outlined below**):

1. **code.zip**: Your data processing, classification and evaluation code (as some combination of .py and .ipynb files).
2. **writeup.pdf**: Your (max 5-page) project write-up as a pdf (details below).

Project write-up

Your team must submit a project write-up that is a maximum of five pages (single-spaced, 11pt font or larger; minimum 0.5 inch margins, an extra page for references/bibliographical content can be used). We highly recommend that students use LaTeX to complete their write-ups. **This first mini-project report has relatively strict requirements, but as the course progresses your project write-ups will become more and more open-ended**. You have some flexibility in how you report your results, but you must adhere to the following structure and minimum requirements:

Abstract (100-250 words) Summarize the project task and your most important findings. For example, include sentences like "In this project we investigated the performance of linear classification models on two benchmark datasets", "We found that the logistic regression approach was achieved worse/better accuracy than naive Bayes and was significantly faster/slower to train."

Introduction (5+ sentences) Summarize the project task, the two datasest, and your most important findings. This should be similar to the abstract but more detailed. You should include background information and citations to relevant work (e.g., other papers analyzing these datasets).

Datasets (5+ sentences) Very briefly describe the and how you processed them. Describe the new features you come up with in detail. Present the exploratory analysis you have done to understand the data, e.g. class distribution.

Results (7+ sentences, possibly with figures or tables) Describe the results of all the experiments mentioned in Task 3 (at a minimum) as well as any other interesting results you find. At a minimum you must report:

1. A discussion of how the logistic regression performance (e.g., convergence speed) depends on the learning rate. (Note: a figure would be an ideal way to report these results).
2. A comparison of the accuracy of naive Bayes and logistic regression on both datasets.
3. Results demonstrating that the feature subset and/or new features you used improved performance.

[1] Ng AY, Jordan MI. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems 2002 (pp. 841-848).

Discussion and Conclusion (5+ sentences) Summarize the key takeaways from the project and possibly directions for future investigation.

Statement of Contributions (1-3 sentences) State the breakdown of the workload across the team members.

Evaluation

The mini-project is out of 100 points, and the evaluation breakdown is as follows:

- Completeness (20 points)
  - Did you submit all the materials?
  - Did you run all the required experiments?
  - Did you follow the guidelines for the project write-up?
- Correctness (40 points)
  - Are your models implemented correctly?
  - Are your reported accuracies close to the reference solutions?
  - Do your proposed features actually improve performance, or do you adequately demonstrate that it was not possible to improve performance?
  - Do you observe the correct trends in the experiments (e.g., comparing learning rates)?
- Writing quality (25 points)
  - Is your report clear and free of grammatical errors and typos?
  - Did you go beyond the bare minimum requirements for the write-up (e.g., by including a discussion of related work in the introduction)?
  - Do you effectively present numerical results (e.g., via tables or figures)?
- Originality / creativity (15 points)
  - Did you go beyond the bare minimum requirements for the experiments? For example, you could investigate different stopping criteria for logistic regression, investigate which features are the most useful (e.g., using correlation metrics), or propose an automated approach to select a good subset of features.
  - **Note:** Simply adding in a random new experiment will not guarantee a high grade on this section! You should be thoughtful and organized in your report.

Final remarks

You are expected to display initiative, creativity, scientific rigour, critical thinking, and good communication skills. You don't need to restrict yourself to the requirements listed above - feel free to go beyond, and explore further.

You can discuss methods and technical issues with members of other teams, but **you cannot share any code or data with other teams.**