Midterm Project 1

Due Dates, Deliverables, and Grading

Check mycourses for most up-to-date due dates.

Due	Activity
check mycourses	Group registrations (mycourses group tab)
check mycourses	Model choice due (spreadsheet link in mycourses)
check mycourses	Code submission due
check mycourses	Presentation

Overview

This is the first midterm project of the course. This project involves reviewing existing literature about a machine learning model (see below), implementing the model "from scratch" (see below the allowed libraries to be used), and conducting experiments using the model on student-chosen data.

After groups are formed and ML model is chosen, groups will work on the implementation of the model according to the requirements listed below.

Pre-project

Make sure you:

- Are in a group by the due date
- Choose a model by the due date

Check mycourses for the due dates.

Implementation requirements

- Code should be in a single Google Colab page

For development purposes, any IDE or environment is allowed, however at time of submission the group need to make sure that the program is organized and submitted as a **single Google Colab page**. You will fill the spreadsheet with the Colab link by the due date.

- Code should be runnable in the Google Colab environment

Before the submission, the group need to make sure that the code will be able to run in the Google Colab environment. Any instructions (e.g. the need to use a kernel with a GPU) need to be present as part of the ipynb file itself.

- Data to train or do inference with the model(s) should be available

Make sure the data used in this project is available publicly in some storage platform (e.g. Google Drive) without the need to logon. The ipynb should provide code to download and uncrompress the data required for the training and inference of the model.

- For implementing your model, you can only use functionality from these libraries: numpy, scipy, pandas, matplotlib, autograd.

Using pytorch, tensorflow, sklearn for you model chosen is not allowed. In the exceptional case where there is a need to use an additional library different than those listed, explicit authorization from the instructor is required. For the baseline model you can use any library/repository.

- Code generation is allowed

If you opt to use it, list the prompts in your submitted Google Colab.

- You should not make your project implementation public

This includes any code repositories. Not following this will be considered academic dishonest conduct, subject to university, department and instructor policy and sanctions.

Data

You are allowed to use any freely accessible dataset. You are also allowed to generate the data that will be used by the model. Datasets must have at least 1000 samples. If you intend to use a dataset with fewer samples than that, get explicit permission from the instructor.

The group is free to choose the task and data that is more appropriate for the experiments with the model chosen. While you can iterate about possible tasks and datasets, you can not change the model after it set in the spreadsheet.

Model choice

The group will choose one of the following listed models. The choice should be done by filling the associated project spreadsheet (linked from mycourses). It is not allowed two groups to implement the same model.

Supervised Models

- Multinomial naive bayes
- Polynomial regression
- Logistic regression
- ID3 Decision Trees

Supervised Representation Learning

- Linear discriminant analysis
- QDA
- Supervised PCA
- Locality Snensitive Hashing

Unsupervised Representation Learning

- NMF
- Kernel PCA
- Isomap
- Multi Dimensional Scaling

Task definition and data requirements

The choice of task should take into consideration the model chosen and the data that will be used to train and perform inference with the model.

If your model uses more than one model (e.g. dimensionality reduction followed by a logistic regression classifier), you are only required to implement one of them and you should indicate the model that you will implement in the spreadsheet. The other model implementation can be obtained from anhy library as long as your ipynb take care of loading it.

Evalution and visualizations

- Choose at least one commonly used metric for evaluation

The group should prefer metrics that are typically use in similar published papers that perform similar ML tasks. The group is encouraged to use more than one metric for evaluations. For instance, if you are performing classification, you might want to use accuracy or F1 score, but it is recommended that you also use precision and recall.

- Perform evaluation on both the training and the testing splits

You need to implement the code to calculate your metrics.

- Choose a baseline

You need to choose at least one baseline, that can be a model that you have not implemented and obtained from an outside library (e.g. sklearn, gensim, etc). The chosen mode can be more or less complex than the chosen model.

- Generate visualizations

Generate graphs showing the performance (using the chose metrics)

- of your model and your baselines
- both at training and testing
- show the tuning (e.g. gridsearch) of the hyperparameters of your model (at least one hyperparameter)

Proposal Presentation and Peer Feedback

Prepare and present the work completed in this project.

- You must use slides.

It is fine to use additional resources such as showing visualizations, animations and demonstrations, but slides are mandatory.

- Every teammember needs to present
- You presentation must be between 8 and 12 minutes long.
- You must use the peer feedback form that will be available in mycourses to evaluate every other team. Each student needs to make one evaluation submission for each presentation watched. E.g. if there are 9 groups, you will submit the evaluation form 8 times.
- You will not evaluate your own group.
- The group that performs the best (based on peer feedback) will get an automatic 100 for the project presentation grade.
- The presentation needs to cover the following topics
    1. Learning task in the context of the data used
    2. Detailed explanation of the learning model
    3. Experiment design, how the split was performed, description of the dataset and preprocessing steps
    4. Results, evaluation metrics, comparisons with the baselines

Grading rubric

Code (50% of project grade)

- Code is in the right ipynb format in a single file
- Code runs on Google Colab without modifications
- Data was able to be downloaded just by running the ipynb cells
- Model chosen was correctly implemented, uses approved dependencies
- Data chosen is of appropriate minimum size
- Data splits were created and correctly used
- Evaluation metric was implemented
- Visualization generated of performance/error of the model training vs testing
- Visualization generated with the model testing performance vs baselines
- Visualization generated of hyperpareter tuning of the model

Presentation (50% of project grade)

- Link is posted in spreadsheet
- Presentation length is/was between 8 and 12 minutes
- All group members presented
- Peer feedback completed
- Peer feedback scores