INF2008 Group Project

Project description

The objective of the project is to let students have a hands-on experience on machine learning application development, this will help the students to have a better understanding on the topics and algorithms learned. Specifically in problem formulation, data collection and processing, data analysis, experiment design, machine learning methods comparison, performance evaluation, and result analysis. Each group has its flexibility to choose its own problem and data.

The recommended workflow of the project will likely consist of the following stages:

1. Problem statement and data collection

Find out an interesting problem to work on. You may refer to the example project in 'Example projects' section below. Those examples are just for your reference, you are not limited to those applications. During the project planning, one important issue is the availability of data. If you decided to work on a problem, you need to first make sure there are data available, or you have the mechanism to collect/label data. For example, you can make use of web crawler application to crawl some data from internet. You should be able to generate a project plan at this stage.

2. Data processing, analysis, and feature extraction

When you have enough data, you may want to perform some data cleaning/filtering, and feature extraction for your problem. For example, the natural language processing (NLP) related data needed to be converted into numeric vectors. You may also want to try out some data visualization techniques to understand and analyse your data.

In this stage, you should also split data for your experiments. In machine learning applications, we usually separate the entire data into 3 parts, training, development, test. The training and development data will be used for model training and parameter tuning. The test set you will not look at during the model training process, it will be used for evaluating the performance. (Think about why?)

3. Experiment design and machine learning algorithm implementation

In this stage, the following questions needed to be addressed: given the problem and data, what would be the most appropriate machine learning method to solve your problem? How do you design the experiments to validate your method, what performance metric is most appropriate for your problem?

You may want to compare **at least two classical machine learning algorithms**[1] to support your decision of algorithm choice. During comparison, except the accuracy, you may also think about practical issues, such as model size, computation speed in training and testing, memory, etc. to justify why you choose the method.

You are allowed to use any **public notebooks, codes** from Kaggle or other open-source platforms. If you choose to use any public notebooks, you should refer to **at least two**

[1] suggested to better use the algorithms covered in the lectures, this provides alignment between what you learn in class and what you apply in the project. Avoid using deep learning techniques, this is disadvantage those without a background in deep learning.

different notebooks and find a way to combine their approaches to improve the overall performance compared to the individual notebooks.

In your slides and video presentation, clearly explain the original methods used in the referenced notebooks and the improvements you implemented. Be sure to **highlight what was original and what you modified or improved.**

4. Experiment results and analysis
In this stage, examine what is the performance of your method. If the results are reasonable? Are you using the best method for the problem, or what are the directions for better performance. You may need to go back to the previous stage several times to re-run your experiments to get better results.

5. Report and presentation
This is the stage to summarize the project, each group needs to come up with:

1) a PowerPoint slide with **maximum 10 slides** with main content **+ one more** slide provides the 1) Youtube link and 2) GitHub or Google Colab link
2) Video presentation **no longer than 10 minutes** with normal speed (create a YouTube link)
3) Source code - GitHub repository or Google colab (create a link with access granted)

Platform and language

The following platforms and languages are a good start for you, but you are not limited to use them:

Google Colab
Python/C/C++
Any machine learning platforms, eg, sklearn, Tensorflow, Pytorch etc

Grouping

We have grouped students according to the sequence of their student IDs for each lab session. Most groups consist of 5 members, while a few have 6 members. Please find your lab and group number at Xsite -> Content -> Group Project to self-enroll.

Xsite->Class activities->Groups-> Self enrollment project groups

The deadline for self-enrollment is Friday Week 1, 11:30PM

Deliverable (per group)

1. Overview

Deliverable	Description	Due (**strictly no extension**)

Project presentation slides (**maximum 10 slides** with main content)	PowerPoint presentation .pptx The PPT should be informative enough about the project details, reader can have the whole picture of the project by only reading this	Friday Week 11, 11:30PM
Video presentation (**no longer than 10 minutes per group**)	Video presentation has each group member	
Source code with user manual	Include the data and source code. The user manual shall describe the details about environment setup, data processing, and training/testing. Readers shall be able to reproduce results by following the user manual	
Peer review	Rating member's contribution. Each group member will receive an email about one week before the deadline	

Peer review: if no peer review is received from the group after the deadline, equal contributions for every member will be assumed.

2. Final deliverable: the final deliverable should be just **project_presentation.pptx** and submit to LMS, please rename it to

project_presentation<XX-XX>.pptx
e.g. if you are in the LAB-P1 and group 1, the filename should be
project_presentation-P1-01.pptx

In this **project presentation slides**

1) **Project main content not exceeding 10 slides**:
The presentation should describe the following aspects: group introduction (who did what), what is the problem, why the problem is important, the details about the data, the toolkit you used, machine learning algorithm adopted and the rational of using those methods, the experiment setup, experiment results and discussion. You may include resource information in the appendix (not counted in 10 pages).

2) The last slide (e.g. slide11) with a YouTube **video presentation link and source code link** with granted access

Video presentation: Record a video presentation to introduce your project orally based on your presentation slides. It should be no longer than 10 minutes. All group members MUST present, but we do not enforce how much each member covers. Please indicate your name in your presentation (e.g., bottom-left of the screen or slides) to facilitate our marking process. Please make sure the video is in normal speed (do not speed up). Make it to a YouTube link.

Source code: Please zip all the source code as well as the start guide (or manual) into ONE (1) zip file and submit the file in LMS Dropbox. System libraries are not required to submit. In case the file is too big, please submit the Github link (make sure it is accessible by instructors).

4. Peer review
The review will be done by filling in an online survey/form. I will email everyone with the link near the submission period. The peer review will be used to facilitate the marking of members in the group. If no peer review is received for the group, an equal contribution will be assumed.

* Fo presentation slides/video presentations, please do not exceed the page/time limits as instructed above. **Penalty applies if the reports exceed the limitation**.

Example projects

1. Classification/prediction/analysis on publicly available data, eg, https://data.gov.sg, some examples for your reference:
    A. HDB resale price prediction: https://data.gov.sg/dataset/resale-flat-prices
    B. Covid trend forecast: https://data.gov.sg/search?q=covid
    C. CF Crime Dataset: https://www.kaggle.com/datasets/odins0n/ucf-crime-dataset
    D. Bank Transaction Dataset for Fraud Detection: https://www.kaggle.com/datasets/valakhorasani/bank-transaction-dataset-for-fraud-detection
    E. Retailrocket recommender system dataset: https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset

2. Natural language processing
    A. Document classification: we deal with a lot of text information every day. The topic classification task is to categorize the article or paragraph into pre-defined categories. For example, you can classify articles according to the topics, you can classify user's comments into positive or negative according to their review. You may find plenty of such dataset on https://www.kaggle.com/
    B. Token tagging is one of the fundamental components in natural language understanding. The objective is to classify tokens (words/phrases) into pre-defined labels. Named entity recognition (NER) is one of the most common tasks, where words/phrases are categorized into entity groups like: names, numbers, locations, currency, dates, company names, etc.
    C. collection of NER data can be found at: https://github.com/juand-r/entity-recognition-datasets
    D. Sentiment classification: https://github.com/jeffprosise/Deep-Learning/blob/master/Sentiment%20Analysis.ipynb

3. Audio/speech classification
    A. Audio classification, spoken digit classification data: https://github.com/soundata/soundata#quick-example

4. Image classification
    A. Handwritten image recognition: Assessment criteria: https://www.kaggle.com/code/manthansolanki/image-classification-with-mnist-dataset/notebook

5. Healthcare application (proposed by Dr Zha Wei, if you are interested in these projects, you may seek his suggestions)
    A. **Background**: Healthcare affordability is one of the key concerns of Singaporeans. Medical waste and abuse from inappropriate claims contribute to escalating costs,

without benefiting patients. The Table of Surgical Procedures (TOSP) is an exhaustive list of procedures for which MediSave / MediShield Life can be claimed One most command inappropriate claim type is that doctor submitting more than one TOSP code where a single TOSP code adequately describes the episode of surgery/ procedure carried out.
Task: Develop an approach that can identify inappropriate TOSP code pairs. For example, doctor submits SL701L (LENS, CATARACT (PAEDIATRIC), EXTRACTION WITH ANTERIOR VITRECTOMY) and SL702L (LENS, CATARACT (PAEDIATRIC), EXTRACTION WITH ANTERIOR VITRECTOMY - BILATERAL) together under one single claim. https://isomer-user-content.by.gov.sg/3/ca783b21-2842-4431-b2f0-3934be261852/table-of-surgical-procedures-(as-of-1-jan-2024).pdf

B. EEG signal detection: https://ieeg-swez.ethz.ch/

6. Practical AI systems (by Dr. Liu Junhua)

Background: Academic studies in Machine Learning largely focuses on proposing new **models** or mechanisms to solve common prediction tasks better (e.g. improving prediction accuracy). When we design ML systems in production (i.e. there are real human using your app!), there are many practical challenges to be overcome, such as cost, efficiency, safety, and many more.

Dr. Liu has over 10 years of experience in building and deploying AI systems to over 30 enterprise clients, from international enterprises to government agencies, to local businesses. As an associate faculty, he has mentored numerous university and polytechnic ML projects which produced well recognized publications in international conference proceedings and journals.

If you are interested in working on practical ML topics that may result in a decent publication or an open-source project that becomes a good record in your portfolio, you may reach out to Dr. Liu (email to: j@forth.ai) to express your interest, with or without a project idea in mind.

Some of the following topics are active work Dr. Liu is working on, either for his startup product or for actual clients. If your project produces good outcome, your work may receive further endorsement from the industry and be used by real users.

Potential Topics:

A. **Finance AI:** Financial AI leverages AI models to automate and enhance financial performance in different aspects, such as financial market forecasting, portfolio optimization, and automated trading. This topic aims to design ML models or pipelines to address practical problems in the Finance domain.
Examples: https://github.com/junhua/awesome-finance-ai-papers

B. **Responsible AI**: Fairness has emerged as a critical concern in terms of both the predictive outputs of ML models. Challenges in fairness often stem from data biases and algorithmic limitations, which can propagate or exacerbate existing inequities. This topic investigates fairness and bias in existing AI models and implement mechanisms to mitigate such bias while maintaining model performance.
Examples: https://github.com/junhua/bgm-han

C. **AI-augmented decision making:** decision making in high-stake scenarios is often done by human experts leveraging their domain expertise and experience to optimize decision quality. However, subjectivity and cognitive biases in the process is often difficult to detect and avoid. This topic focusses on designing AI-augmented, bias-aware decision making process in a real world setting, where fairness and consistency is well taken care of.
Examples: https://arxiv.org/pdf/2411.17374

D. **Customer Support AI**: Customer support AI in production remains primitive despite recent advancement in Generative AI. Even today when you call in a CS hotline or chat with a widget, you still go through a fixed list of options in hope of finding a solution (which more often end up talking to a human support). In this topic, we aim to tackle challenging sub-tasks, such as intent recognition, knowledge base management, response generation and escalation path optimization. Examples:
[1] https://arxiv.org/pdf/2411.14252
[2] https://arxiv.org/pdf/2411.12307

7. Practical AI systems (by Dr. Rishabh)

A. **Accurate electricity consumption forecasting** is paramount for effective energy management. By anticipating energy demand, suppliers can optimize distribution, minimize waste, and prevent system overload. However, traditional forecasting methods often face limitations in terms of accuracy and scalability. Therefore, a robust and efficient approach to predicting electricity usage is essential.
Paper Link: https://www.e3s-conferences.org/articles/e3sconf/pdf/2023/28/e3sconf_icmed-icmpc2023_01048.pdf
Sample dataset: https://www.kaggle.com/code/nageshsingh/predict-electricity-consumption (Students should find other datasets as well)

B. **Prediction of student performance using machine learning models.** Found one highly cited papers for the same: https://slejournal.springeropen.com/articles/10.1186/s40561-022-00192-z

Assessment criteria

	Excellent	Good	Average	Fail to meet expectations
Presentation slides quality (40%) **Slides**	The presentation slides are clear and logical. Excellent coverage and structure with clarity of explanation and details. Methods, analysis, results, conclusions are clearly stated. Implications of results and analysis are discussed.	The presentation slides generally clear, mostly good report with explanation and details, some minor errors and ambiguities. Clear description of methods and some discussion of the results with some conclusions.	The presentation slides are unclear with less structure. Report organization is not well thought out. Partial explanation of methods and results, some errors and ambiguities. Some discussion of the results, little or no conclusions.	The presentation slides are very confusing and unclear. Brief and minimal report with little explanation and details. Nothing much done.

	Excellent	Good	Average	Fail to meet expectations
Project quality, novelty, and implementation (40%) **video + ppt + source code**	Interesting, innovative, accurate and useful analysis and features. Appropriate & sufficient data sources are collected/used for the data analytics tasks. Significant implementation to data analytics accuracy, appropriate choice of toolkits/API, algorithms and or complete implementation of modules. Excellent coding quality.	Minimum suitable datasets are collected/used for the analysis. Appropriate and useful analysis tasks are performed. Good implementation to support features/use cases, appropriate choice of toolkits/API for the working demo. Well tested and working. Good coding quality.	Very few features or use cases supported. Insufficient or inappropriate data sources are used. Implementations are not always justified. The analysis did not come out with much useful information. Minor implementation for the sake of analysis tasks, and limited features, may not work completely. Codes with some bad practices.	Total copy of existing idea, or nothing much done. Few features or use cases. Total copy of existing code, very little implementation, or nothing much done. Codes are wrongly implemented.
Experiment evaluation and insight discussion (20%) **video + ppt + source code**	Experiments are well designed and documented. Good interpretation of results and conclusions. Results and conclusions are clearly stated with further implications. Limitations may be stated too. Very interesting insight was identified. Evaluation and insights are clearly documented. Evaluation results and conclusions clearly stated and justified with suitable reasons and implications, solutions are designed and stated.	Appropriate choice of experiments. Results and conclusions are clearly stated, some further implications given. Some good interpretations and conclusions. Interesting insight was identified. Evaluation and insights are documented. Results and conclusions are stated, some good reasoning and conclusions with solutions.	Experiments are not always clearly documented. Some partially correct interpretations, results and conclusions. Limited insight was identified. Evaluation and insight are not completely documented. Not all results and conclusions are stated, partial conclusions and solutions.	Hasty, poor or very basic experiments done. Experiments not clearly documented. Few, wrong or no conclusions. Hasty, poor or very basic evaluations done. Evaluation not clearly documented. Few, wrong or no insights.

*Extra bonus marks will be given for innovative algorithms which can outperform the state-of-the-art algorithms via benchmarking and experimental comparison. To do so, you may need to use the same performance measure and dataset to make a fair comparison.

Your project may also be selected for publication and presentation in an international/national conference. This will add good value to your CVs. We will work together with your group on the submission of the paper. SIT is going to pay for your conference registration and publication fee.

Late Submission

A penalty of 20% per day for each deliverable will be imposed for late submission unless an extension has been granted prior to the submission date. Request for extension will be granted on a case-by-case basis. Any work submitted more than 4 days after the submission date will not be accepted and no mark will be awarded.

Plagiarism

SIT's policy on copying does not allow you to copy software as well as your assessment solutions from another person. It is not acceptable to copy another person's work. It is the students’ responsibility to guarantee that their assessment solutions are their own work. Meanwhile, you must also ensure that others don’t obtain access to your work. Where such plagiarism is detected, both assessments involved will receive ZERO mark.