CHE 4230 Advanced Process Control Systems – Spring 2025 Semester Project (4230)

Introduction

The project will focus on data analysis and machine learning, encouraging students to explore unsupervised and supervised learning concepts using a dataset of their choice. Students will learn to apply dimensionality reduction and clustering algorithms to find meaningful insights from their data as well as apply classification and/or regression methods to make predictions and assess their performance. The goal is to allow flexibility in data selection while maintaining consistency in learning outcomes and deliverables. Some example data could be sports, stocks, genetics, etc. Of course, you can choose chemical plant data as well. If you do not want to pick your own data, you can use the Tennessee Eastman (TE) data and follow that example project.

Part 1: Project Setup and Planning

1. **Deliverables**:
    - A GitHub repository with an organized directory structure for source code, analyses, and documentation. (See Appendix 1 for example repository setup).
    - A 2-page project plan outlining:
        - The selected dataset and its context (e.g., industry, domain).
        - Why the choses dataset is suitable for unsupervised and supervised tasks.
        - Basic exploratory data analysis (EDA) to justify data selection.
        - Proposed tools/methods for data analysis in Parts 2 and 3.
        - Tentative internal deadlines for each project part.

Part 2: Unsupervised Learning (Data Analysis and Pattern Recognition)

1. **Objectives**:
    - Explore patterns using unsupervised methods.
    - Test and compare various dimensionality reduction (≥5) and clustering (≥3) techniques.
2. **Deliverables**:
    - Preprocessing code tailored to the dataset.
    - Implementation of DR and clustering methods.
    - Annotated Jupyter Notebook with results and insights.
    - Relate clustering results to real-world interpretability based on the dataset domain.
    - A 1-page summary discussing findings and comparisons.

Part 3: Supervised Learning (Classification and/ or Regression)

1. **Objectives**:
   - Develop and evaluate two models; either classification models or regression models depending on your use case.
     1. Some classification models could be Artificial Neural Networks (ANN) and Support Vector Machines (SVM) [not limited to these].
     2. Some regression models could be Random Forest or Gradient Boosting [not limited to these].
     3. Packages like scikit-learn have algorithms that you can use.
   - Compare the effectiveness of the models using different preprocessing techniques and hyperparameter tuning.
2. **Deliverables**:
   - Preprocessing code specific to the supervised task.
   - Implementations of two predictive models.
   - Annotated Jupyter Notebook showcasing experiments and analyses.
   - A 1-page summary of findings, including a performance comparison between the models.

Part 4: Research Evaluation (only if you took the 7000 version of the course)

1. **Objectives**:
   - Choose a scholarly paper from platforms like Google Scholar or other academic databases.
   - The paper should use supervised or unsupervised learning techniques. Ideally, select a paper that uses both approaches, if possible.
   - Key points to address:
     1. Problem definition: What is the paper addressing? Why is the problem significant?
     2. Dataset: describe the dataset(s) used. How were the data prepared for modeling?
     3. Machine learning techniques: What supervised/unsupervised learning algorithms were applied? Why were these methods chosen? Were any modifications or innovations introduced to the standard techniques?
     4. Results and Evaluation: Summarize the key results. What metrics were used for evaluation? How do the results compare to baseline methods or prior work (if applicable).
     5. Discussion: What are the strengths and limitations of the approach? Can you identify and areas for improvement or future work? What are some practical implications or applications of the work? Is there anything that you thought was exceptionally interesting?

2. **Deliverables**:
    - 5-10-minute PowerPoint presentation slides
    - Try to have clear slide structure that is not too wordy. Focus on the main details.

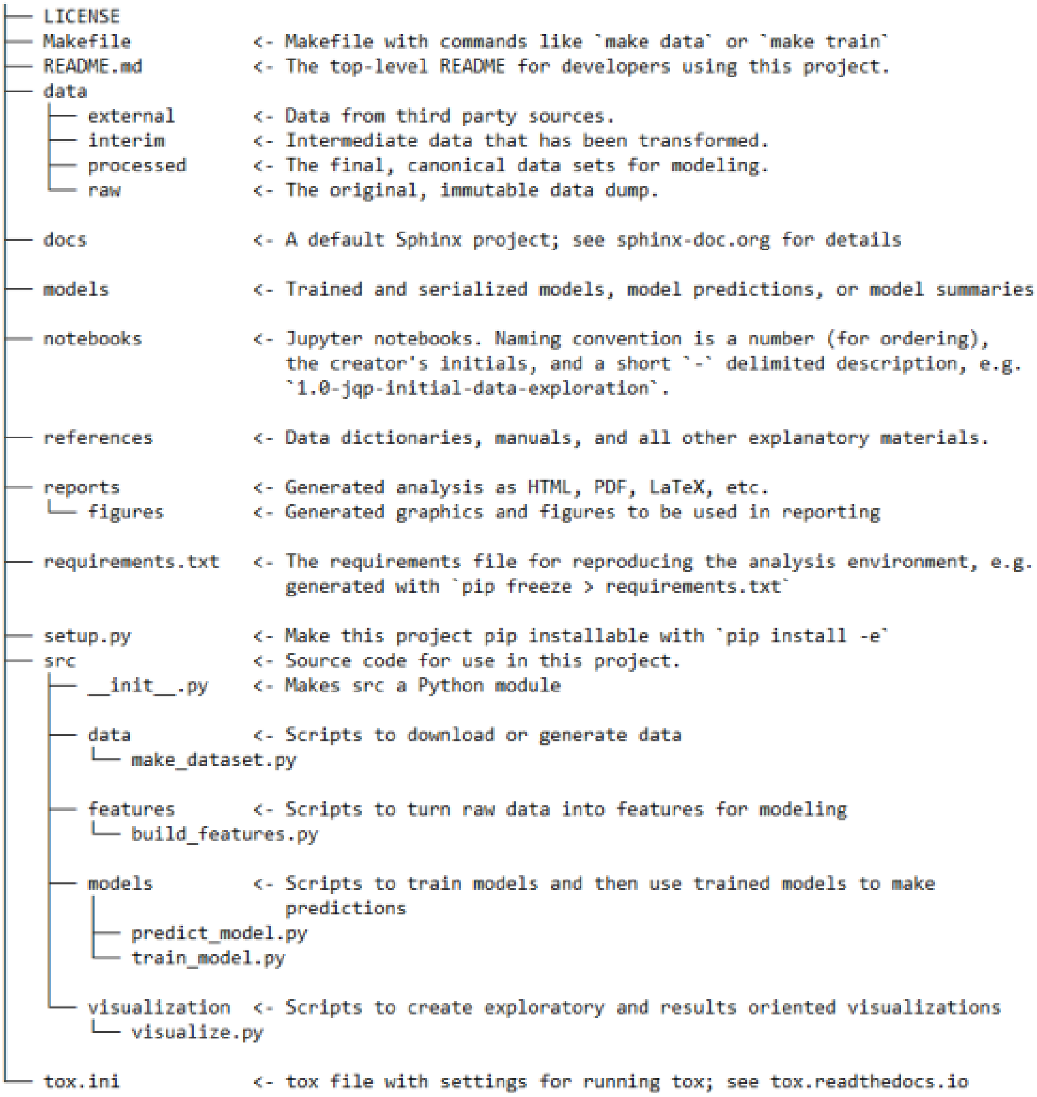
Grading and Evaluation

- **Preliminary Report**: Includes Part 1 and a brief justification for the dataset choice.
    - GitHub repository (50%).
    - Project plan (50%).
- **Final Report**:
    - Deliverables for Parts 2 and 3: 80%.
    - Performance and insight quality: 20%. (20% of best model accuracy)
- **Presentation (only if you took the 7000 version of the course)**:
    - Presentation will consist of 5% of the Final Report grade.

Due Dates

1. **Preliminary Report- February 14 (just include Part 1)**
2. **Final Report- March 14 (Parts 2 and 3)**
3. **Presentations (only if you took the 7000 version of the course)- February 27**

Appendix 1. Sample of directory structure



```
├── LICENSE
├── Makefile           <- Makefile with commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
│
├── docs               <- A default Sphinx project; see sphinx-doc.org for details
│
├── models             <- Trained and serialized models, model predictions, or model summaries
│
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
│
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
│
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
│
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
│
├── setup.py           <- Make this project pip installable with `pip install -e`
├── src                <- Source code for use in this project.
│   ├── __init__.py    <- Makes src a Python module
│   │
│   ├── data           <- Scripts to download or generate data
│   │   └── make_dataset.py
│   │
│   ├── features       <- Scripts to turn raw data into features for modeling
│   │   └── build_features.py
│   │
│   ├── models         <- Scripts to train models and then use trained models to make
│   │   │                 predictions
│   │   ├── predict_model.py
│   │   └── train_model.py
│   │
│   └── visualization  <- Scripts to create exploratory and results oriented visualizations
│       └── visualize.py
│
└── tox.ini            <- tox file with settings for running tox; see tox.readthedocs.io
```