2025Spring DSAA2011 Course Project

Teaching Team of DSAA2011

Hong Kong University of Science and Technology (Guangzhou)

Introduction

This project is designed to give students hands-on experience in applying machine learning techniques to real-world datasets sourced. Students will complete a series of specified tasks—including data visualization, clustering, model training, and performance evaluation—while also engaging in open-ended exploration to encourage creativity and deeper analysis. The project aims to simulate solving a practical problem using a dataset, mirroring real-world machine learning workflows.

You are required to complete a series of specified tasks—including data visualization, clustering, model training, and performance evaluation—while also engaging in open-ended exploration to encourage creativity and deeper analysis.
All projects will be completed in groups of **2–3 students**.

Project Guideline

What makes a **good** project? Here are a few considerations:

- Clear outcome to predict
- Some techniques learned from class should do something interesting, e.g., linear regression, feature engineering, Gradient boosting and etc.

Instructions:

1. Select and apply **at least two** algorithms.
2. Apply the techniques to the selected dataset and assess their effects.
3. Document the process, including code, visualization, and analysis.

Deliverables:

1. Description of the chosen techniques and implementation.
2. Results (e.g., improved metrics, visualizations) and insights gained.

Dataset

Each group can select a dataset below for analysis.

- **Adult Income Dataset**: Contains 48,842 instances from the 1994 U.S. Census, with 14 features (e.g., age, education, occupation, marital status, hours worked per week) and a binary target indicating whether income exceeds $50K/year.
- **Human Activity Recognition Using Smartphones Dataset**: Contains 10,299 instances of sensor data (accelerometer and gyroscope) from smartphones worn by 30 subjects performing 6 activities (e.g., walking, sitting, standing). Features include 561 time and frequency domain variables extracted from raw signals..

- **Dry Bean Dataset**: Contains 13,611 instances of dry bean measurements with 16 features (e.g., area, perimeter, major axis length, eccentricity) and 7 classes representing different bean varieties (e.g., Seker, Barbunya, Bombay).

Project Timeline

All project deadlines are at 11:59pm GMT.

- **April 20, 2025**: Grouping initialization. Build a group with 1-2 friends, select a dataset and inform us. You should join only one group. Each dataset can be selected a maximum of 10 times, with allocations granted on a first-come, first-served basis.

  How: Leader of each group must write ONE response to the Discussion topic 'Teams: Group Information and Selected Dataset' in Canvas. Write down the list of group members and the selected dataset.
- **April 27, 2025**: Grouping confirmation. Students who are not in any group will be randomly grouped. The groupID of each group will be released.
- **May 18, 2025 (last day before exam week)**: Submission due.

Mandatory Tasks

You are expected to complete all following tasks using the selected dataset.

1. Data Preprocessing

- Instructions
  1. Examine and Handle missing values (e.g., fill the missing value, add a corresponding label).
  2. Handle non-numeric values (e.g. one-hot encoding, Boolean indicator).
  3. Further processing (e.g. standardize features).
- Deliverables
  1. Brief description of (a) the applied method(s) and (b) observed patterns of dataset or insights.

2. Data Visualization

- Objective: Visualize high-dimensional data in a 2D and/or 3D space using t-SNE.
- Instructions
  2. Create a scatter plot of the resulting embedding, coloring points by class labels if applicable.
  3. Analyze the visualization to identify patterns or clusters.
- Deliverables
  1. Plot of the t-SNE projection.
  2. Brief discussion of observed patterns or insights.

3. Clustering Analysis

- **Objective**: Group data points into clusters and evaluate the clustering quality.
- Instructions
    1. Select **at least two** suitable clustering algorithm (e.g., K-means, hierarchical clustering).
    2. Apply the algorithms to the preprocessed dataset.
    3. Evaluate the results using multiple metrics.
    4. Visualize the clusters (e.g., scatter plot with cluster labels).
    5. Determine the best clustering results and justify it.
- Deliverables
    1. Description of the chosen algorithms and why they were selected.
    2. Evaluation and interpretation of clustering results.
    3. Visualization of the clusters.
    4. Comparison of algorithm performances.

4. Prediction: Training and Testing

- **Objective**: Train supervised learning algorithms and assess its performance.
- Train a simple machine learning model and assess its performance.
- Instructions
    1. Choose a classification target (e.g. classification of a value).
    2. Choose **at least two** simple model classes (e.g., decision tree, logistic regression).
    3. Split the dataset into training (e.g., 70%) and testing (e.g., 30%) sets.
    4. Train the model classes on the training set.
    5. Test the trained model on the the training set, testing set and the entire set.
- Deliverables
    1. Description of the chosen classification target, model classes and why they were selected.
    2. Description of training process.
    3. Visualization the results (e.g. the decision boundary).
    4. Evaluation of prediction results on the training set, testing set and the entire set individually, including generating confusion matrices (e.g., using matplotlib or seaborn).
    5. Interpretation of prediction results on the training set, testing set and the entire set.
    6. Comparison of algorithm performances.

5. Evaluation and Choice of Prediction Model

- **Objective**: Analyze and improve the models' performance.
- Instructions
    1. Calculate metrics such as accuracy, precision, recall, and F1-score for each model trained in *Part 3* (using confusion matrix).
    2. Draw ROC and calculate AUC for each model class.
    3. Improve each model via validation.

4. Interpret the results to assess each model's strengths, weaknesses and possible improvements (e.g., determining whether the model overfits).

- Deliverables
  1. Calculated metrics, e.g., AUC and plotted ROC.
  2. Description and interpretation of validation results.
  3. Further discussion (100-200 words) of model performance based on metrics, ROC, AUC and so on.

Open-ended Exploration

You are expected to explore more machine learning methods to deepen your understanding of machine learning and further improve the developed algorithms. Creativity and experimentation are encouraged.

Suggested Techniques:

- **Model improvement**: Improve the model by investigating the impact of model complexity, error metric, regularizer (e.g. polynomial regression, Ridge regression).
- **Model Comparison**: Compare at least three different model classes (e.g., SVM, random forest, neural networks) using cross-validation.
- **Feature Engineering and selection**: Create new features (e.g., polynomial features) or select a subset (e.g., using feature importance) and evaluate their impact.
- **Hyperparameter Tuning**: Optimize model parameters using grid search or random search.

Submission Guideline

You should submit a zip file containing the following files

- `report_groupID_dataset.pdf`: A project report in PDF format, summarizing your work and findings.
- `project_groupID_dataset.ipynb`: A Jupyter notebook containing all your code, visualizations, and explanatory text.
- `requirements_groupID_dataset.txt`: A text file listing all Python packages and their versions used in your project.
- (Optional) `data/`: A folder that contains the data you analyze.

The zip file should be named using this convention: **groupID_dataset.zip**, e.g., `01_datasetA.zip`

Report Format (report.pdf)

The report is a comprehensive document that presents your project work in a clear and structured manner. It must follow the following structure:

- Structure
  1. **Introduction**: Briefly highlight the findings and main results of your report.
  2. **Mandatory Tasks**: Present results from tasks such as t-SNE projection, clustering analysis, model training, and confusion matrix analysis.

3. **Open-ended Exploration**: Discuss open-ended analysis (e.g., feature engineering, model selection) you performed.
4. **Conclusion**: Summarize your findings and insights.
5. **References**: List all external resources cited in your report.
6. **Credit**: Contribution of each group member and GenAI tools (e.g., 10% of report writing is polished by Doubao).

- **Format**: You must format your submission using this LaTeX style file. Do not use the "preprint" option. You may consider to use 'Overleaf' for collaboration.
- **Length**: Aim for 5-10 pages, excluding references, ensuring conciseness while covering all required content.
- **Originality**: All work must be your own. Cite any external resources (e.g., code, tutorials) properly to avoid plagiarism.

Jupyter Notebook (project.ipynb)

The Jupyter notebook is the core of your technical submission and should include all code and analysis. Follow these requirements:

- **Content**: Include all code for the specified tasks (data preprocessing, t-SNE, clustering, model training, etc.) and your open-ended exploration.
- Documentation
  - Add **comments** within the code to explain each step clearly.
  - Use **markdown cells** to provide context, describe your approach, and embed visualizations (e.g., t-SNE plots, confusion matrices).
- **Visualizations**: Generate and display all required plots within the notebook, ensuring they are clearly labeled (e.g., with titles, axes labels, and legends).
- **Reproducibility**: The notebook should execute from start to finish without errors in a fresh environment using the packages listed in requirements.txt.

Requirements File (requirements.txt)

This file ensures that your code can be run consistently by instructors. Include:

- A list of all Python packages (e.g., scikit-learn, pandas, matplotlib) and their exact versions used in your project.
- Generate this file easily by running the following command in your Python environment:

  text

  `pip freeze > requirements.txt`

This allows instructors to recreate your environment accurately.

Grading Components

The project will be graded out of 100 points, based on the following:

- **Correct Implementation (30%)**: Accuracy and completeness of the specified tasks.
- **Analysis and Interpretation (30%)**: Depth and quality of insights provided for each task.

- **Open-ended Exploration (30%)**: Creativity, effectiveness, and analysis of additional techniques.
- **Report Quality (10%)**: Clarity, organization, and professionalism of the report.
- **Contribution**: Contribution of GenAI tools and each group member will additionally impact the grade.

Important Notes

- **Late Submissions**: Late submissions may incur a penalty of 10% for all group members per day. Any submission delayed by less than 24 hours will still be counted as a full 24-hour period.
- **Usage of GenAI tools**: The use of GenAI tools such us LLMs (<30%) is permitted only for support tasks such as asking question, code debugging, generating visualizations, or drafting text for the report. However, all core machine learning implementations (e.g., t-SNE, clustering, model training) and analysis must be your own work. If you use such tools, clearly disclose their usage in the **References**: A section of your report, specifying the tool and how it was applied (e.g., "Used Grok by xAI to generate initial t-SNE code snippet, which was then modified"). Failure to disclose this may be considered a violation of academic integrity.
- **Questions**: If you have any doubts about the submission process, contact the teaching assistants well before the deadline.