Second Assignment

Clustering, Neural Networks, Dimensionality Reduction

Course: Machine Learning
Instructor: *Fabio Patrizi*

Due Date: *05/01/2026*

Objective

The goal of this project is to apply in practice and analyze the following approaches: K-means for Clustering; Neural Networks, including CNNs, for Regression and Classification; Principal Component Analysis (PCA) and Autoencoders for Dimensionality Reduction.

Assignment Description

Part 1: Data Selection

For each of the following problems, choose a suitable dataset from the UCI Machine Learning Repository or Kaggle: clustering, multi-class classification, regression. For classification, choose an image-based dataset (you can select). The dataset must be of appropriate size (at least 1,000 data points).

Part 2: Data Preprocessing

Perform appropriate preprocessing steps:

- Handle missing or noisy data.
- Normalize or standardize features.
- Split data into training, validation, and test sets.
- Fix a reasonable dimensionality for each dataset and reduce the data to that by applying both PCA and Autoencoders. When applied to the same dataset, both approaches must reduce the input to the same dimensionality.

Part 3: Model Implementation and Training

Define and train the following models (and set the hyper-parameters via cross-validation):

1. K-Means for clustering;
2. FNN for regression;
3. CNN for classification.

Each model must be trained on the original (if manageable) and the two reduced datasets, as resulting from both the Autoencoder and PCA, for a total of 3 trained models per task.

Part 4: Evaluation

Choose appropriate metrics to evaluate each model. For each of them, compare the results on the original and reduced data obtained with PCA and Autoencoder.

Evaluate each model using the following metrics, when applicable:

- Accuracy, Precision, Recall, F1 Score
- Confusion Matrix
- ROC and AUC (binary and multiclass)
- Training vs. validation performance
- **Optional**: Computational cost and training time

Part 5: Comparative Analysis

Include a 1–3 page analysis discussing:

- Which models performed best and why.
- How model assumptions influence performance.
- Observations on overfitting trade-off.
- Visualizations: learning curves, decision boundaries, feature importance.

Deliverables

1. **Technical Report (PDF) prepared in LaTex**:
   - Introduction and motivation
   - Dataset description
   - Methodology and models
   - Results and analysis
   - Conclusion
2. **Google Colab Notebook** with clean, commented code, to reproduce all the experiments presented in the technical report.

Suggested Tools

There is no need to implement everything from scratch, you can use the following tools:

- Python: `NumPy`, `Pandas`, `Matplotlib`, `scikit-learn`, ...
- Jupyter Notebook or Google Colab
- LLMs: they are great to guide you through the project, **but** (there is a big **but**) you should not use them black-box without understanding what they are doing and why. Most importantly, they will not be allowed at your written exam, so make sure to use this assignment as an opportunity to learn by doing.

Evaluation Criteria

This is an open ended project. There is no wrong way to set it up, as long as you perform the tasks required and submit it on time. So feel free to be imaginative and creative, play with the models that we have studied in class and test them on real life datasets!

Submission Details

Hard Deadline: 23:59 (CET), 5th of January 2026.
Submission: Use Google Classroom
Team Formation: You can decide to submit the project as a team of at most *two* participants.