Project 1

The aim of the project is to implement **Cyclic Coordinate Descent (CCD)** algorithm for parameter estimation in **regularized logistic regression with l1 (lasso) penalty** and compare it with standard logistic regression model without regularization.

Task 1

The aim of the first task is to collect and prepare data sets for conducting experiments.

- Find **4 different real datasets** corresponding to classification problem with binary class variable containing numerical variables.
  - You can use repositories: https://archive.ics.uci.edu/, https://www.openml.org/ or other sources.
  - Non-standard, interesting datasets will be appreciated. You can convert multi-class datasets to binary datasets by combining classes (for example treat the majority class as positive and assign the remaining observations to the negative class.).
  - Please choose the datasets with large number of features, the number of variables should be at least 50% of the number of observations. If it is not, you can add dummy variables that are copies of the original variables with permuted values.
  - Prepare datasets to run logistic regression algorithms. This includes:
    - filling in missing values,
    - removing collinear variables,
- Generate **Synthetic dataset** using the following procedure (p,n,d,g are treated as parameters which will vary in the experiments):
  - Generate binary class variable (Y=0 or Y=1) from Bernoulli distribution with class prior probability p.
  - Generate feature vector X, such that X|Y=0 follows d-dimensional multivariate normal distribution with mean (0,…,0) and covariance matrix S with $S[i,j] = g^{|i-j|}$, whereas X|Y=1 follows d-dimensional multivariate normal distribution with mean (1,1/2,1/3,…1/d) and covariance matrix $S[i,j] = g^{|i-j|}$.
  - Generate n observations using the above steps.

Task 2

The aim of the second task is to implement regularized logistic regression using algorithm CCD[1] (call the method **LogRegCCD**). The implementation description can be found in the article https://www.jstatsoft.org/article/view/v033i01 The most relevant description is in section 3. However, the preceding sections also contain descriptions necessary to perform the implementation, in particular formulas (4), (5) and (10). It is not necessary to use all the speed-up tricks described in the article, although speeding up the method will count as a plus.

- The input of the method: training data (features X_train and labels y_train), validation data (features X_valid and labels y_valid).
- The optimal value of the lambda parameter should be selected by optimizing the appropriate measure on the validation set. The user should be able to choose the following measures:

[1] Using implementations available on the web or LLM generated code is not allowed, the idea is to write your own functions.

recall, precision, F-measure, balanced accuracy (for a threshold of 0.5), area under the ROC curve, area under the sensitivity-precision curve.

- The following methods should be implemented:
    - fit(X_train,y_train),
    - validate(X_valid, y_valid, measure),
    - predict_proba(X_test),
    - plot(measure, ...), which produces plot showing how the given evaluation measure changes with lambda,
    - plot_coefficients(...), which produces plot showing the coefficient values as function of lambda parameter.

Using implementations available on the web is not allowed, the idea is to write your own functions.

Task 3

The aim of this task is to conduct experiments in which we will evaluate the performance of the implemented method and compare it with the standard implementation of logistic regression without regularization.
As the baseline, use the sklearn implementation:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

(**LogisticRegression** with penalty='None' and default value of the remaining parameters).

- Analyze how parameters n,p,d, g affect the performance of LogisticRegression and LogRegCCD methods based on synthetic data. As an evaluation measure, consider ROC AUC and Balanced Accuracy (for a threshold of 0.5).

Compare:

- the performance of the two methods on real datasets using ROC AUC, Recall-Precision AUC, F-measure and Balanced Accuracy as the evaluation measures.
- coefficient values obtained in these two methods

Final grade

The final grade of project 1 you can get 20 points and they are divided into three components:

- Report – **12 points**
- Code – **6 points**
- Presentation in the form of an interview with the lecturer at the project class – **2 points**

Requirements for reports

Every point described below should be included in separate section. **Maximal length of report is 6 pages A4 (title page and refernces are not included in the limit).** Report should include:

- **Methodology.**
    - Selection and generation of datasets.
    - Details about algorithm implementation and applied optimizations
- **Discussion about correctness of the LogRegCCD algorithm.**
  Suggested approach to address this point:

    - Performance of the algorithm at lambda=0
    - Likelihood function values and coefficient values depending on iteration
    - Comparison with ready implementation of logistic regression with L1 penalty
- **Impact of dataset parameters: n.p,d,g on the performance of LogRegCCD algorithm.**
- **Benchmark of LogRegCCD with LogisticRegression algorithm.**
  Suggested approach to address this point:
    - Performance of algorithms regarding different metrics
    - Values of coefficients obtained in these two methods.

Requirements for code

- Instructions to run algorithm (README)
- Script, notebook to run the algorithm for new data
- Documentation of the functions - **docstrings**
- Readability of code and use of good practices for granularity of functions
- Saving all files (plots, tables) used for the report in separate files

Additional remarks:

- The projects are implemented in teams of 3 students. Teams can be formed between project groups, but each team will be assigned to one class term and instructor
- The projects are implemented in Python or R.

Organization of work

- Project consultations are held every two weeks. Since project classes are not mandatory, the team reports attendance at consultations in the Consultations.xlsx form, and in addition, we ask for a legal message to the corresponding instructor. Consultation attendance should be reported by the preceding Wednesday by 8 pm.

- The teams' composition should be entered in the Teams.xlsx file. The preferred instructor should also be completed. Teams can be completed until 03.03.2025. After this date we will make available the final assignment of teams to instructors and class dates.

- Schedule and topics for Project 1 consultations:
    - 20/27.02* and 6/13.03* - selection of datasets and implementation of the algorithm
    - 20/27.03* - analysis of results

  *Date appropriate to the project group

Final submission

- Please prepare zip file (name of the file: Surname1_Surname2_Surname3.zip) including 3 folders: code, presentation, report. Please upload your solution using the task assigned in the MS Teams channel.
- Deadlines: 31.03.2025.
- Presentations:
    - group 1: 03.04.2025,
    - group 2: 03.04.2025,

  - group3: 10.04.2025,
  - group4: 10.04.2025.
- If you have any questions, please send us an e-mail: katarzyna.woznica@pw.edu.pl, adam.majczyk.stud@pw.edu.pl, dawid.pludowski.stud@pw.edu.pl