Sarah Schröder, Alexander Schulz — Bielefeld University

Introduction to Machine Learning (WS 2025/26)
Project

Released: Thursday, 08.01.2026.
Code Submission: Thursday, 22.01.2026. Submit as executable python script(s).
Code Presentation: Monday, 26.01.2026 in the tutorials. The tutors will arrange time slots for each group. The code will be graded based on the presentation and submission to Moodle.
Report submission: Sunday, 01.02.2026. You receive the grading for the report in Moodle.
Grading: Code and report are graded separately. In the report both the report quality (structure, clarity of writing, plots) and the content (how well methods are described, depth of your analyses) will be graded.

- You may hand-in your report in German as well as in English, but be consistent. We will not substract any points for errors regarding language as long as your report is understandable.
- We provide a LaTeX template `report_template.tex` which might help you.
- There is a Q&A tutorial for this project on Monday, 19.01.2026.
- Usage of LLMs: We expect the code and report to be **your** work. Using LLMs for smaller tasks such as language polishing, looking up syntax or overcoming writer's block is fine. Copy-pasting larger pieces of text without further adaptations or understanding is not.
- If you use LLM generated code or code from the internet be transparent about it and reference the sources.
- If you have any questions, please ask your tutor or write an email to intromachlearn@techfak.uni-bielefeld.de.

Work on the tasks described below and write a **full-text** report on **max. six** pages (excluding figures and tables). Your report should contain

- a description of the data sets used (especially in real-world datasets with meaningful features!),
- descriptions of models and techniques used (especially those aspects relevant for your experiments and analysis of results),
- a documentation of your experimental set-up (What choices (e.g. metrics, preprocessing, hyperparameters) did you make? Why?),
- your results and an analysis (can you explain your results? how does this relate to the properties of the methods used or the principles of machine learning? what are the takeaways?),
- and, of course, a short introduction and conclusion.

When writing your report, please make sure that your descriptions are complete and precise. Based on your text we should be able to reproduce your pipeline and your results. Besides, you will have quite a lot of results. Consider one part of the task it to present them in a consist way. For example, you can consider visualizing your results in plots or creating tables.

When putting together your report pay attention to the structure. This is very important, as it is hard grasp work described in a badly structured report. We will also grade the report quality apart from content, such as structure, clarity of descriptions and analysis, plot quality and completeness of report (consider the bullett points above).

Hint: When creating plots using matplotlib, you can save the current figure by `plt.savefig('path/to/file.eps', format='eps')`. If you are using LaTeX for the first time `www.tablesgenerator.com` might be a helpful resource for easily creating clean looking tables.

1 Random Forest and Feature Importances

(*40 Points*)

Grading: Coding (14 points) , Report Content (16 points) and Report Quality (10 points) .

(a) Train and evaluate a random forest classifier on `dataset_rf.npz`. Compare its performance to the other classifiers you know from the lecture (Naive Bayes, Logistic Regression, kNN).

(b) Visualize the data by plotting each combination of two features. Analyze the feature importances of the random forest with respect to the data. Which type of feature relevance computation is best here (impurity or permutation based)? Evaluate the classifiers on a suitable subset of the features.

(c) Compare the OOB score of the random forest classfier with the cross-validation score for $n \in \{5, 10, 25, 50, 100, 200\}$ trees. Make sure the OOB and CV scores are comparable by computing them on the same amount of data.

2 Loan Applications

(*40 Points*)

Grading: Coding (14 points) , Report Content (16 points) and Report Quality (10 points) .

For this task, consider the credit loan dataset (`credit_data_train.csv` and `credit_data_test.csv`). Using Machine Learning to make decisions on loan applications poses ethical challenges. Thus, in this task we consider the fairness at the example of binary gender. A simple measure for fairness could be the difference of some performance measure between men and women.

(a) Preprocess the dataset, considering categorical features and other apsects

(b) Evaluate the performance of classifiers known from the lecture (Random Forest, kNN, Logistic Regression and Naive Bayes, no hyperparameter tuning necessary). Select the two best performing / most reasonable classifiers for the following subtasks.

(c) Consider the 'Gender' feature to analyze the classifiers performance with respect to men and women separately. Which performance measure would be most relevant given the task at hand? What do you observe? What are the implications or possible reasons?

(d) Think of at least two (simple) strategies that could improve the classifiers both in terms of performance and the performance gap between genders. Discuss your results under those two aspects.

Hint: We recommend pandas to load and preprocess this dataset.

3 Clustering

(*40 Points*)

Grading: Coding (15 points) , Report Content (15 points) and Report Quality (10 points) .

In this task, evalaute clustering algorithms on BERT embeddings from the news_groups dataset. The dataset contains online posts about different topics (it's the same dataset from Programming Sheet 2, except the embeddings have been normalized for your convenience). You can load it from `news_data_normalized.npz` (fields: text, y, emb, labels).

(a) Train and evaluate DBSCAN without using the ground truth labels. In particular think about the hyperparameters (epsilon and min_samples) and how to choose them. How does this influence the number of clusters? Report at least one internal metric.

(b) Report an external metric and visualize the ground truth labels and predictions of DBSCAN. Use UMAP (see info below) to project the embeddings into 2D. Analyze how the clusters found by DBSCAN relate to the ground truth classes.

(c) Train, evaluate and plot KMeans in the same way for $n \in \{4, 6, 8\}$ clusters. Compare the performance (based on the metrics and plots) to DBSCAN. How do the clusters relate to the ground truth classes for different values of n.

UMAP:

UMAP (Uniform Manifold Approximation and Projection) is a technique for dimensionality reduction to 2 or 3 dimensions, such that we can plot and interpret the dataspace. While the low dimensional projection cannot capture the entire complexity of high-dimensional data, UMAP preserves both local (near-by) and global (far-away) structure comparably well. Since UMAP was not covered in the lecture, you do not need to understand the details or describe it in your report. However, if you want to find out more, check out the UMAP documentation.

You can install UMAP with

```
pip install umap-learn
```

and in your python script project some high dimensional data X to 2D by calling:

```
map_reducer = umap.UMAP(n_neighbors=15, min_dist=1.0, metric='euclidean',
    n_components=2, random_state=42)
X_2d = umap_reducer.fit_transform(X)
```