

Project Outline and Report

Your project report at the end of the quarter should contain the following sections. The descriptions of these sections below should give you a general idea of how to organize and focus your time in your project.

Report Title + Authors: e.g., "An investigation of classification methods for CIFAR-10", plus the team-member names and student IDs

Please be sure to use the section headings and numbers below, in this order, in your report

1. **Summary** a short 2-4 sentence summary of your project, identifying the dataset(s) you used, what classification methods you investigated, and 1 or 2 main conclusions from your investigations
2. **Data Description:** describe the dataset(s) that you used in your project. Describe what dataset exploration you did, e.g., visualization, generation of summary statistics. Include at least 1 figure in this section. Try to find and read at least one published paper that has used this dataset for research in the past (you can use the paperswithcode Website to find papers).
3. **Classifiers:** list the classification algorithms/methods that you used in your project. For each classification method provide a brief 1 or 2 sentence description of the method (e.g., imagine your report is being read by a computer scientist who is not familiar with the details of different classifiers, but would like a high-level summary of the characteristics of each). Mention what the hyperparameters are for each method (if any) and for each hyperparameter the range of values (or settings, or architectures) that you investigated in your experiments. Mention also what software you used for each method, e.g., scikit-learn or some other package like PyTorch (for more advanced neural networks).
4. **Experimental Setup:** describe briefly how you conducted your experimental methodology. What metrics did you look at? e.g., just classification accuracy (or error), or other metrics (such as precision/recall for binary classifiers). How did you partition up your data? we recommend that at the start of your project you set aside a test dataset that is only used once at the end of your project for final evaluation of models (this will be a realistic test of your method), e.g., 20% randomly selected examples, or the specified test set if the dataset has such. The rest of the data (e.g., the remaining 80%) can be partitioned into a training and validation set (e.g., 75% of the 80% (i.e., 60% of the total) for training and 25% of the 80% (i.e., 20% of the total) for validation, where the validation set can be used for hyperparameter tuning. Be clear on how you selected hyperparameters: feel free to use pseudocode for example to describe precisely what you did. In reporting machine learning experiments it is important that your experiments are reproducible (for others to

independently validate and recreate what you did), and reproducibility implies clear documentation.

5. **Experimental Results:** summarize the results that you obtained on the test data for each of your classifiers and for each of the metrics that you used, and write a brief interpretation of the overall results. Provide the results as a table or as a figure (e.g., bar-chart). Feel free to add additional information, such as comparing performance of each model on test data to its performance on training data. Feel free to also add additional experimental results (such as how accuracy for certain classifiers varied if they were provided with less training data (i.e., "learning curves"), or how accuracy varied as a function of regularization) - this level of detail can go in an Appendix if you don't have space in the main report. You could also analyze if classifiers make the same errors: e.g., if classifier A has an error rate of 5% and classifier B has an error rate of 10%, are all (or most) of classifier A's error included in classifier B's errors?
6. **Insights:** write a brief summary of what you learned by doing this project. For example did any of the results surprise you? You could also do an "error analysis" of the results by looking at the examples where the classifier made an error: are the examples that the classifier misclassifies also hard for humans to classify? and if so, what makes these examples difficult? Feel free to speculate on what you think the strengths and weaknesses of using any of the machine learning methods (that you investigated) on a more realistic version of the dataset(s) you worked with, e.g., putting the classifier on a real camera for object recognition, or using it in a real-world medical setting.
7. **Contributions:** for each team member provide a 1 to 2 sentence description of how that team member contributed to the project (e.g., it could be in terms of contributions to the different aspects of your project as described in the section headings above). While there will inevitably be a fair bit of overlap in terms of what team members work on (and that is fine) please try to identify at least 1 or 2 aspects of the project, for each team member, where they did the most work relative to the rest of the team.

Your team will produce a single document as a report (to be submitted as a PDF file), approximately 4 pages long with sections for each of the headings described above.

Important: the goal of your project is not to produce the most accurate state-of-the-art classifier in the world for a particular dataset. Instead the goal is for you to learn about developing, evaluating, and comparing machine learning methods and to be able to systematically evaluate the strengths and weaknesses of different approaches. A well-executed project with a well-written report, where it's clear that the team learned a lot by doing the project, even where the final test accuracy is not state-of-the-art, can get a higher score than a project that has high accuracy but where you didn't learn much. In this sense, the sections where you describe your experimental results and your insights are particularly important in your report.

Classifiers: for your classifiers we require that at a minimum you include the following methods in your experiments:

- kNN, logistic, and feedforward neural networks (with at least 1 hidden layer)
- and in addition, at least 1 other type of classifier (other than nearest-centroid, which is too simple): e.g., you could use a decision tree type of classifier (random forests, gradient-boosted trees, etc), and/or a support vector machine classifier, and/or a more specialized form of neural network (e.g., a convolutional model for image datasets or a recurrent neural network for text classification). You are required to have 1 classifier (it can be any kind of classifier, beyond the 3 required types), but can have more if you wish and are free to explore classifiers that we did not discuss in class. If you have time you could also investigate if combinations (i.e., ensembles) of your classifiers produce better results than any classifier alone.
- Note that if you use a complicated classifier (e.g., beyond those we discussed in class) in your project then we expect that you have an understanding of how it works: not necessarily all the details, but you should be aware of the basic principles of how the classifier works, what the hyperparameters are for the classifier and what they mean, what the default settings are for these settings in an open-source packages you call, etc. As a computer scientist or data scientist working with machine learning tools you want to not just use "black-box" code in your scripts, but have a general understanding of what these black-boxes are actually doing.

You are welcome to discuss your projects on Ed with other teams but should not share any code: e.g., you can post (and answer) questions about a dataset or classification algorithm or some Python library; or mentioning ideas or posting links that you found useful for your project. We will setup tags on Ed for project discussions (with subtags for specific datasets, etc)

You are free to use any **external software packages** that you find useful. These include, but are not limited to Scikit-learn, Pytorch or Tensorflow (for neural nets), Pandas (for data processing), etc.