

# Project 3 on Machine Learning, deadline December 10

## Data Analysis and Machine Learning FYS-STK3155/FYS4155

Department of Physics, University of Oslo, Norway

Nov 12, 2018

### Premises for project 3

For project 3, you can propose own data sets that relate to your research interests or just use existing data sets from say

1. [Kaggle](#)
2. The [University of California at Irvine \(UCI\)](#) with its machine learning repository
3. The credit card data set from [UCI](#) is also interesting and links to a [recent scientific article](#).

The approach to the analysis of these new data sets should follow to a large extent what you did in projects 1 and 2. That is:

1. Whether you end up with a regression or a classification problem, you should employ at least two of the methods we have discussed among **linear regression (including Ridge and Lasso)**, **Logistic Regression**, **Neural Networks**, **Support Vector Machines** and **Decision Trees and Random Forests**. If you wish to venture into **convolutional neural networks** or **recurrent neural networks**, or extensions of neural networks, feel free to do so.
2. For project 3, you should feel free to use your own codes from projects 1 and 2, eventually write your own for SVMs and/or Decision trees and random forests' or use the available functionality of **scikit-learn**, **tensorflow**, etc.
3. The estimates you used and tested in projects 1 and 2 should also be included, that is the  $R^2$ -score, **MSE**, cross-validation and/or bootstrap.

4. If possible, you should link the data sets with existing research and analyses thereof. Scientific articles which have used Machine Learning algorithms to analyze the data are highly welcome. Perhaps you can improve previous analyses and even publish a new article?
5. A critical assessment of the methods with ditto perspectives and recommendations is also something you need to include.

All in all, the report should follow the same pattern with abstract, introduction, methods, code, results, conclusions etc as the two previous projects.

We propose also an alternative to the above. This is a project on using machine learning methods (neural networks mainly) to the solution of ordinary differential equations and partial differential equations.

This is a field with a large interest recently, spanning from studies of turbulence in meteorology to the solution of quantum mechanical systems.

**Part a): Text to come.**

## Introduction to numerical projects

Here follows a brief recipe and recommendation on how to write a report for each project.

- Give a short description of the nature of the problem and the eventual numerical methods you have used.
- Describe the algorithm you have used and/or developed. Here you may find it convenient to use pseudocoding. In many cases you can describe the algorithm in the program itself.
- Include the source code of your program. Comment your program properly.
- If possible, try to find analytic solutions, or known limits in order to test your program when developing the code.
- Include your results either in figure form or in a table. Remember to label your results. All tables and figures should have relevant captions and labels on the axes.
- Try to evaluate the reliability and numerical stability/precision of your results. If possible, include a qualitative and/or quantitative discussion of the numerical stability, eventual loss of precision etc.
- Try to give an interpretation of your results in your answers to the problems.
- Critique: if possible include your comments and reflections about the exercise, whether you felt you learnt something, ideas for improvements and other thoughts you've made when solving the exercise. We wish to keep this course at the interactive level and your comments can help us improve it.

- Try to establish a practice where you log your work at the computerlab. You may find such a logbook very handy at later stages in your work, especially when you don't properly remember what a previous test version of your program did. Here you could also record the time spent on solving the exercise, various algorithms you may have tested or other topics which you feel worthy of mentioning.

## Format for electronic delivery of report and programs

The preferred format for the report is a PDF file. You can also use DOC or postscript formats or as an ipython notebook file. As programming language we prefer that you choose between C/C++, Fortran2008 or Python. The following prescription should be followed when preparing the report:

- Use Devilry to hand in your projects, log in at <http://devilry.ifi.uio.no> with your normal UiO username and password and choose either 'fysstk3155' or 'fysstk4155'. There you can load up the files within the deadline.
- Upload **only** the report file! For the source code file(s) you have developed please provide us with your link to your github domain. The report file should include all of your discussions and a list of the codes you have developed. Do not include library files which are available at the course homepage, unless you have made specific changes to them.
- In your git repository, please include a folder which contains selected results. These can be in the form of output from your code for a selected set of runs and input parameters.
- In this and all later projects, you should include tests (for example unit tests) of your code(s).
- Comments from us on your projects, approval or not, corrections to be made etc can be found under your Devilry domain and are only visible to you and the teachers of the course.

Finally, we encourage you to collaborate. Optimal working groups consist of 2-3 students. You can then hand in a common report.

## Software and needed installations

If you have Python installed (we recommend Python3) and you feel pretty familiar with installing different packages, we recommend that you install the following Python packages via **pip** as

1. pip install numpy scipy matplotlib ipython scikit-learn tensorflow sympy pandas pillow

For Python3, replace **pip** with **pip3**.

See below for a discussion of **tensorflow** and **scikit-learn**.

For OSX users we recommend also, after having installed Xcode, to install **brew**. Brew allows for a seamless installation of additional software via for example

1. brew install python3

For Linux users, with its variety of distributions like for example the widely popular Ubuntu distribution you can use **pip** as well and simply install Python as

1. sudo apt-get install python3 (or python for python2.7)

etc etc.

If you don't want to install various Python packages with their dependencies separately, we recommend two widely used distributions which set up all relevant dependencies for Python, namely

1. [Anaconda](#) Anaconda is an open source distribution of the Python and R programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment. Package versions are managed by the package management system **conda**
2. [Enthought canopy](#) is a Python distribution for scientific and analytic computing distribution and analysis environment, available for free and under a commercial license.

Popular software packages written in Python for ML are

- [Scikit-learn](#),
- [Tensorflow](#),
- [PyTorch](#) and
- [Keras](#).

These are all freely available at their respective GitHub sites. They encompass communities of developers in the thousands or more. And the number of code developers and contributors keeps increasing.