UiT

THE ARCTIC UNIVERSITY OF NORWAY

Assignment 1. Due: Sunday 08.09.2024 23:59

FYS-2021 Assignment

Department of Physics and Technology
Faculty of Science and Technology

Guidelines

The assignment is mandatory. You get a pass/fail grade. If you fail it you can't come to the exam and you fail the entire course. To pass, you need to:

- write a short report (pdf file) where you provide your answers to the questions asked, and upload it on Canvas,
- Succeed in the task of making the machine learn (even if it makes a poor classification at the end)
- upload (or push) your code to your Github repository. You may upload the file directly using the browser if git does not work on your computer.

Additional important guidelines

Learning to write a scientific report is an important skill that many of the courses at the Faculty of Science and Technology, including this one, aim to improve. Therefore any question that you answer should be contained within the report of this assignment. Answers outside of the written report, (e.g. in the comment of the code, or within a Jupyter Notebook), will not be considered as a part of your answer of the problem. You can structure your report by having a separate (sub)section with the answer for each question. The report should contain text, figures, excerpts of code and anything that helps you answering the questions. The report and code should be your own individual work. Remember to cite all sources.

Make sure your report shows that you understand what you are doing. More specifically, it is important to elaborate your answers such that essential theory, equations, and intuition is included in your answers. However, your answers should still remain concise and stay focused on the core problem, e.g. there is no need to derive or prove an equation unless the problem asks you to.

Problems that ask for numeric values or plots should include these in the answer of the report. The code should be commented in such a way that any person with programming knowledge should be able to understand how the program works. Like your report, the code must be your own individual work.

You are permitted to use standard built-in functions and/or packages (e.g. numpy, pandas and matplotlib in Python) for reading the data and basic calculations. However: make sure that the packages you use do not over simplify your implementation (using Scikit-Learn is not permitted!). Of course, all

implementations asked for in the problems should be your own work.

Submission

For your final submission, upload the report as a single `.pdf` file on Canvas under 'Assignments' > 'Mandatory assignment 1' by the announced submission due. Make sure you have your name written on the report and a link to your Github repository so that we can find your code. The naming convention for the file should be `assignment1_firstname_lastname.pdf` Upload your code on Github and write a short description of it in the readme file. You may use some text of your report in this readme file. You may upload you report there as well if you wish.

Plagiarism

Plagiarism is a serious academic offense and will not be tolerated. Always ensure that you provide proper attribution for any work, ideas, or concepts that are not originally yours. Should the Canvas plagiarism detection system flag your submission, your report will not be considered for assessment.

Resources

All datasets required to answer the exercises can be found in the Canvas room for the course.

Problem 1

Introduction. We have an internal debate in the Machine Learning group about music and we need some help from you. For a few of our favorite songs, we disagree on the genre it should be classified in. Some of us think they are inspired by Pop and should be tagged "Pop", but a minority think that the influence comes from Classical music and should be classified in this category instead. In order to end this dispute, we ask you to design a machine that classify automatically a song, given its features.

For this assignment you will be working with the spotify-dataset. You can also find the link in Canvas. Download the data from this page, uncompress it, you should have a file named `SpotifyFeatures.csv`.

Preprocessing. An important skill for a machine learning engineer is the ability to process raw data. In this problem you will be asked to select samples from different genres, as well as labeling them according to class.

(1a) Load the `SpotifyFeatures.csv` file and report the number of samples (songs) as well as the number of features (song properties) in the dataset. Hint: you may use the Python module Pandas and its function `read_csv`.

(1b) You will be working with samples from two genres namely 'Pop' and 'Classical'. Retrieve all samples belonging to the two genres and create labels for the samples i.e: 'Pop' = 1, 'Classical' = 0. Report how many samples belongs to the two classes. Working with all features is not always the best solution since it increases the computational cost and some of them may be useless for the task. For this dataset you should be able to separate the two classes by using two features, namely 'liveness' and 'loudness'.

(1c) From the reduced dataset, make 2 numpy arrays. The first array will be the matrix with songs along the rows and songs' features ("liveness" and "loudness") as columns. This will be the input of our machine learning method. The second array will the vector with the songs' genre (labels or target we want to learn). Create a training and test set by splitting the dataset. Use an 80% 20% split between the training and test set. Split the data per class so that you keep the same class distribution in the training and test set.

(1d) [Bonus.] Plot the samples on the liveness vs loudness plane, with a different color for each class. From the plot, will the classification be an easy task? why?

Problem 2

Machine learning. We are now ready for the machine learning part. The task is to classify the songs as "Pop" or "Classical". The classification method we have seen so far is called Logistic regression. It is very powerful and we are going to use it for this assignment. Let us see if you are a good teacher for the machine.

(2a) Implement your own logistic discrimination classifier and use the training data to train the classifier. You should use stochastic gradient descent and implement it in Python. Plot the training error as a function of epochs, and report the accuracy on the training set. Try different learning rates for the gradient descent and explain what you observe for these different values. Optional, it may help the learning process if the data is shuffled (songs are fed to the classifier in random order).

(2b) Test your trained logistic discrimination classifier using the test set. Report the accuracy on the test set. Is there a significant difference between the accuracy on the training and test set? If so what might that indicate. [Optional] Inform (or brag) about your test accuracy score on Discord. It is nice to seen how other students perform while working on the assignment.

(2c) [Bonus] Extract the learned parameters from your logistic regression and use them to draw the linear line separating the data on the plot you made in question (1d). This may help you understand why your classifier is performing well or not.

Problem 3

Results. In this part we will see how smart is your creation. In order to trust it, we need a machine which does not make too many errors.

(3a) Using the classification results from the test set in problem 2, create a confusion matrix for the classification. Report the confusion matrix.

(3b) You should now have two evaluation metrics for the performance of the classifier on the test set; accuracy and the confusion matrix. What information does the confusion matrix give you that the accuracy score does not?

(3c) [Bonus.] Which songs are difficult to classify? Could you suggest some Classical songs that a Pop fan would like? (a good song could influence positively the mood of the evaluator)