University of St. Gallen

Institute of Computer Science

Fundamentals of Computer Science (3,125), HS2019
T. Huber, C. Niklaus, B. Bermeitinger, J. Sales, S. Handschuh
thomas.huber@unisg.ch

Assignment 8: Machine Learning Basics

Deadline: Dec 03, 2019; 12:00 CET

Introduction In this assignment you will learn how to build a simple text classifier with `scikit-learn`. You will work with real data and get firsthand experience by working on a real world example. Your task is to build and evaluate three different classifiers that have a certain minimum performance. For this you will use the metrics you have already used last week: Accuracy, precision, recall and F1-measure.

Rationale It is always a good idea to get practical experience to reinforce the theoretical constructs you have learned in the lecture. Furthermore building a classifier without evaluating how good it is makes little sense. You will be able to apply the evaluation metrics in other areas as well. This week you will use `scikit-learn` to build your classifiers instead of doing it by hand.

Prerequisites Create a Google account if you do not already have. Then upload the notebook for this assignment on Google Colab: https://colab.research.google.com/.

- Click `File` in the top-right menu
- Click `Upload notebook...`
- In the menu select the template file from the .zip file that you downloaded from Canvas `assignment7_template.ipynb`
- Click the arrow on the right and upload the other files from the .zip file in the `Files` tab

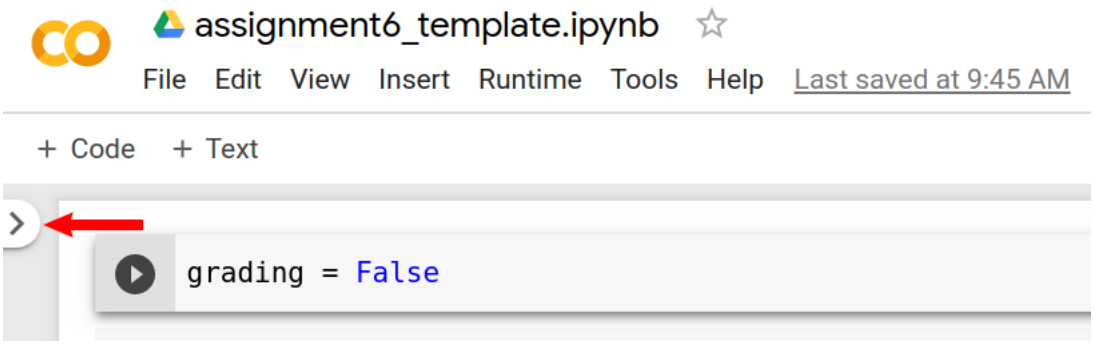

Figure 1: How to open the sidebar with the `Files` tab

For this assignment, you **ONLY** need to edit the notebook!
To run your code, click into one of the cells (the blocks of text that you can edit) and then click the *Run* button on the side of the cell.
When the code in a cell depends on a previous cell (for example one cell has a class *Villain*) then you need to run the previous cell before running the cell that depends on it! This also applies when you change code in a cell, you need to run that cell again and also any cells that depend on it (for example when they use a function defined in that cell).
You can run multiple cells by using the `Runtime` menu at the top.

(1) Text Classification (5 Points)

In this task you will build three classifiers and evaluate them.

(a) Implement the `load_questions` method.

(b) Implement the `build_dataframe_q` method.

(c) Implement the `create_train_test_set` method.

(d) Implement the `gridsearch` method.

(e) Implement the `evaluate_classifier` method.

(f) Implement the `train_classifier` method.

(2) Evaluation (5 Points)

(a) Optimize the classifier parameters in the `build_pa` method so that all metrics are above 96%. Use your `gridsearch` method for this.

(b) Optimize the classifier parameters in the `build_mnb` method so that all metrics are above 79%. Use your `gridsearch` method for this.

(c) Optimize the classifier parameters in the `build_svc` method so that all metrics are above 96%. Use your `gridsearch` method for this.

Hand-in Instructions Create a file `README.txt` that includes 1.) answers to the questions, 2.) any reference that you used to complete this assignment, 3.) pitfalls you encountered and 4.) short explanations of your solution if necessary. Fill in your name and student ID in the comment header of files you edited.

Download your notebook by clicking `File` at the top and then select `Download .ipynb` and `Download .py`. Include both of these files in your `zip` file.

Compress the whole folder with the Python files and the `README.txt` to a `zip` file named "`assignment6.zip`". Upload the zip file to your exercise group's Course page on Canvas. See Figure.2 for the list of files that should be included in the zip file.

```
assignment8.zip
├── README.txt
├── assignment8_template.ipynb
├── assignment8_template.py
└── train_5500.label.txt
```

Figure 2: The required files for submission.