# Final project

Submit Assignment

**Due** 12 Jun by 23:59 **Points** 0 **Submitting** a file upload **File types** pdf

In the final project, we will work toward actually creating a Fake News predictor. This will build on the work you have done in the Milestone, combined with the topics covered in the lectures in this second block of the course.

## 1. Know your data

Your milestone was primarily about getting to know your data and representing it in a reasonable way. The first part of your final project is to summarize the main findings from this process (possibly incorporating feedback that you got in Peergrade):

1. Describe how you ended up representing the FakeNewsCorpus dataset (for instance by describing your ER diagram). Argue for why you chose this design.
2. Did you discover any inherent problems with the data while working with it?
3. Report key properties of the data set - for instance through statistics or visualization. If you use non-trivial SQL queries to extract these properties, please describe them.
4. What were your experiences with scraping your assigned fragment of the "Politics and Conflict" section of the Wikinews site?

To go further on the work you started with the milestone, we ask you take the following steps:

5. Create a relational database schema to represent the dataset you scraped from the "Politics and Conflict" section of the Wikinews site and import the data you scraped into this schema. Document your schema design in an ER diagram and briefly discuss how you dealt with the metadata in this source.
6. Use SQL to report basic statistics on this additional data source, e.g., number of articles or distribution over dates.
7. Now that you have two different sources in the database, corresponding to the FakeNewsCorpus and to the Wikinews fragment you scraped, create a view that integrates the article information from the two sources. How do you map the different metadata from the sources into a common schema? NOTE: You need at a minimum to create a view schema that will suffice for the modeling task below, though you may include more metadata in the view if possible.

Finally, conclude by specifying how you will use the data to train a Fake News predictor:

8. Specify which data you will be using to train and test the models in the remaining part of this project. Does it makes sense to include the Wikinews data or will you limit yourself to (a subset of) FakeNewsCorpus. Argue why.
9. In this project, we will consider fake news detection as a binary classification problem. Find a good way to aggregate the many output classes of FakeNewsCorpus into 2 classes (FAKE/REAL). Argue why this is a reasonable choice.

## 2. Establish a baseline

The next step is to create one or more reasonable baselines for your Fake News predictor. These should be simple, either in the sense of using only a very simple set of features, or using a very simple model (or both). You will be using these to benchmark your more advanced models against later.

1. Start by considering only features extracted from the main text (content) field. Choose one or more simple baseline models, train them, and report their accuracies. Describe why these are reasonable baselines.
2. Consider whether it would make sense to include meta-data features as well. If so, which ones, and why? If relevant, report the performance when including these additional features and compare it to the first baselines. Discuss whether these results match your expectations.

For the remainder of the project, we will limit ourselves to main-text data only (i.e. no meta-data). This makes it easier to do the cross-domain experiment in question 4 (which does not have the same set of meta-data fields).

## 3. Create a Fake News predictor

Create the best Fake News predictor that you can come up with. Argue for why you chose this approach, and discuss potential alternatives. Quantify its performance against your baseline(s).

## 4. Performance beyond the original dataset

Now, we will test how well the model works beyond the dataset that you described in question 1.

1. We have set up a friendly competition between the groups. The idea is that we provide a dataset *without* labels (CSV format, two columns: ID,text), and that you all use your model to try to predict the labels. You will then upload a file with the ID and the labels (CSV format: two columns: ID, "REAL" or "FAKE"). We will then compare your predictions against the true labels and create an online leaderboard where you can see your rank compared to the other groups. The leaderboard is hosted as a Kaggle competition accessible **here**

(https://www.kaggle.com/t/71a99d7c10144ea597901cb7d24e0bdd) . You can also find the test data set there. Please don't try to reverse-engineer the source of the data we provide, in order to download and train on it (we will be able to tell).
2. In order to allow you to play around cross-domain performance locally as well, try the same exercise on the LIAR dataset (https://www.cs.ucsb.edu/~william/data/liar_dataset.zip (https://www.cs.ucsb.edu/~william/data/liar_dataset.zip) ), where you know the labels, and can thus immediately calculate the performance.
3. Compare the results of these two experiments to the results you obtained in question 3. Report both your LIAR results and the leaderboard results as part of your report. Remember to test the performance of your baseline model as well.

## 5. Discussion

Conclude your report by discussing the results you obtained

1. Explain the discrepancy between the performance on your test set and on the LIAR set and leaderboard. If relevant, use visualizations or statistics to point out differences in the datasets.
2. Conclude with describing overall lessons learned during the project, for instance considering questions like: Does the discrepancy between performance on different data sets surprise you? What can be done to improve the performance of Fake News prediction? Will further progress be driven primarily by better models or by better data? Is it even a solvable problem?

## Practicalities

Although we have previously worked primarily with jupyter notebooks, we have (for technical reasons regarding the marking process) decided that the final report should be handed in as a .pdf file (you are very welcome to use jupyter notebooks during development, and convert them to .pdf in the end - there are several tools for this). Your report should be no longer than 12000 characters (about 3.5 pages of pure text, in a reasonable font size). In addition to raw text, you can add tables, figures and code up to a maximum of 6 pages in total (you are allowed to provide appendices as well, but we won't guarantee that we will consider them). Please structure your report so that the section numbers correspond to the question numbers. Since you only have very limited space, you don’t have to write an introduction.

Importantly, if you distribute tasks such that some in the group are primarily responsible for particular parts of the report, please explicitly state so in the report (using percentages to specify contribution from each member). If no such annotation is provided, we will assume that all members in the group contributed equally to all parts of the report. Finally, please state the exam numbers of all members of the group on the front page of the report.

~~**The final project must be handed in through Digital Exam.** Digital Exam has support for groups, but it requires that one member from each group creates the group, and then invites the other members of the group to join in. I strongly recommend that you create these groups well before the project deadline, so we can deal with any technical problems early on. Please let us know (by starting a discussion in Absalon) if something is not working.~~

**The final project must be handed in through Absalon** (the exam office has asked us to make this change - see announcement). Make sure that you submit as a group. In order to do this, you will need to create your group under the "People" section. I strongly recommend that you create these groups well before the project deadline, so we can deal with any technical problems early on. Please let us know (by starting a discussion in Absalon) if something is not working.