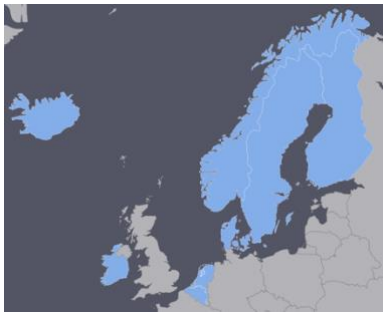
Kaggle Project: Classification of Tweets of Politicians from Northern Europe



Please read all of the guidelines carefully before submitting the lab. ☺ There are ***100 points*** in total. **You can work alone or in a group of two in this project.**

Due date: Monday, November 25, 11:59 PM. Late submissions will be penalized by 5 points / day. No submissions will be accepted two days after the deadline.[1]

Deliverables:

1) The code of the project in **.ipynb** format (one file)
2) The lab report written with ***LaTeX*** and exported in **.pdf** format (one file)

Guidelines – Before You Start

1) **Please do not post any of your code or solutions online. This is a Kaggle competition.**
2) Five high-ranked teams will receive a bonus of 0.5% if they choose to present their methods and classification strategy in class.
3) You will be using the **`Python`** programming language for this project. You need to write your codes in an empty **.ipynb** file.
4) Make sure that you provide many comments to describe your code and the variables that you created.
5) Please use the ***IEEE*** journal template on ***overleaf.com***. Here is the link: https://www.overleaf.com/latex/templates/preparation-of-papers-for-ieee-sponsored-conferences-and-symposia/zfnqfzzzxghk
   To be able to work on ***overleaf.com***, you will need to register first (you can also compile your ***LaTeX*** file locally.)
6) For some of the code, you may need to do a little bit of "*Googling*" or review the documentation.

What is the Twitter Data?

This dataset consists of all tweets posted by politicians of seven different Northern European countries: Belgium, Denmark, Iceland, Ireland, Netherlands, Norway, and Sweden. You will have:

- Tweets posted by politicians of seven Northern European countries mentioned above[2]
- Each country is associated with a different number of tweets. The **`test`** set consists of %20 of the tweets originating from each country (the remaining 80% is the **`training`** set).[3]

[1] **Part II of the analysis needs to be submitted by the deadline (no late submission will be accepted for Part II).**
[2] Earliest tweet is dated December 12, 2008, and the latest tweet is dated January 1, 2023.
[3] The observations in both the **`training`** and the **`test`** datasets are randomly ordered.

- In total, there are 407,223 tweets in the **`training`** set (and 101,808 tweets in the **`test`** set). This makes a total of 509,031 tweets.

Kaggle Project – Data Dictionary

The data has been provided in the assignment folder online (**training_data.csv** and **test_data.csv**). Open the CSV files and take a look at them before starting. Individual features (columns) of the dataset have been described below:

hashtags: The list of hashtags included in the tweet

full_text: The text of tweet (including emojis, htmls, hashtags)

in_reply_to_screen_name: The Twitter screen name of the user the owner of the tweet is replying to (if any)

country_user: Country of the owner of the tweet

pol_spec_user: Political view of the owner of the tweet (found only on the **`training`** dataset)[4]

Id: An index number associated with tweets (found only on the **`test`** dataset)

Before you start with the questions below, create a new and empty folder called **kaggle_project**. Call the file where you will write your code **kaggle_project**, as well.

Part I: Descriptive Analysis (20 points)

In this part of the analysis, you will be exploring some introductory NLP (natural language processing) techniques to better understand the data. Use the **`training`** dataset for the descriptive analysis.

[Important note: Please answer all questions in **Section A** and **Section B**.]

Section A (10 points):

For all questions below, please use the training dataset.

a) Create a **table** that contains information on minimum, average, median, and maximum for the following: tweet length (#characters and #words) (`text` column), hashtag length (#characters and #words) (`hashtags` column) (Add your table to the report.) (**5 points**) (**2.5 points** for graduate students)

b) Find the top ten most commonly used hashtags (`hashtags` column) in each country separately. Then, create pie charts (one pie chart per country) which show the distribution of these ten most commonly used hashtags for each country. Do you observe any patterns? What are the meanings / interpretations of the hashtags you have identified? Write your findings in the report. (Add the pie charts to the report.) (**5 points**) (**2.5 points** for graduate students)

c) Create a stacked bar chart (one stacked bar per country) that shows the percentage of political views associated with each country. [Create normalized bars to show percentages: minimum

[4] There are four political view categories: 'Left', 'Center', 'Right', and 'Independent'.

should be 0, maximum should be 1 (or 0% and 100%)]. Interpret your findings. Add your findings and the graphs to the report. (**5 points**) (**2.5 points** for graduate students)

d) Create a stacked bar chart that shows the distribution of genders by country. [Create normalized bars to show percentages: minimum should be 0, maximum should be 1 (or 0% and 100%)]. Interpret your findings. Add your findings and the graphs to the report. (**5 points**) (**2.5 points** for graduate students)

Section B (10 points):

For all questions below, please use the training dataset.

a) Write a ***'text cleaner'*** function that does the following in the ***full_text*** column: (i) remove stopwords[5], (ii) remove all words that are shorter than 3 characters, (iii) remove all links (starting with *http*), (iv) remove emojis, (v) remove punctuation. Attach the code you wrote to the ***lemmatizer.py*** file in the project folder. Run the lemmatizer function and create 'cleaned and lemmatized' version of ***text*** column. (You can name the new column as ***text_clean***). After the cleaning, expand the table you have created in Section A) by calculating minimum, average, median, and maximum for the newly created ***text_clean*** column (#characters and #words). (**5 points**)

b) Using the code in the following link[6], perform **LDA** (i) and **Non-negative Matrix Factorization** (ii) for topic analysis. Please use the ***text_clean*** column you have created above. Set the number of clusters/topics to 10 (ten) and extract the topics in an unsupervised manner. Adjust any parameters as you see fit. Analyze the results. Compare the results of both models. Interpret your findings and add your findings to the report. (**5 points**)

Part II: Model Creation and Prediction (50 points)

Please do not post any of your code or solutions online.

This part of the analysis needs to be submitted by the deadline (no late submission will be accepted).

Please use the dataset provided to you. [We should be able to run your code with the original datasets and the additional external datasets you provide.] You cannot use any other Twitter data. You can use (non-Twitter) external datasets.

For this part of the analysis, you will need to train a model that classifies the tweets in your training dataset according to '**pol_spec_user'** labels[7], report the **Accuracy** of your best model (i), and the **confusion matrix** (ii) that you will create. **You will mainly be graded on the Accuracy of your model** (more information provided below). Some guidelines (please also review the information shared through lectures):

[5] For more information: https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
[6] https://scikit-learn.org/stable/auto_examples/applications/plot_topics_extraction_with_nmf_lda.html
[7] In other words, you will need to identify the political view of the owners of the tweets found in the `test` dataset.

- Percentages of tweets in the **`training`** dataset associated with different political views are 42.87% (Left), 25.75% (Center), 31.21% (Right), and 0.2% (Independent). Using this distribution, you can set ~42.87% as your benchmark for **Accuracy** (in other words, if you predict all tweets as 'Left', your accuracy will be close to 42.87% - and, 42.87% should be the minimum accuracy you are aiming to achieve).
- You can use **any** classification model (including, but not limited to, logistic regression, decision trees, neural networks etc.)
- Your model should run on a laptop in a reasonable amount of time (in a few hours at a maximum) for grading purposes.
- You can use **any** feature engineering method to transform your dataset, such as:
    - Dimensionality reduction methods such as PCA, t-SNE, spectral embedding
    - Logarithmic, polynomial, and other transformations
    - Different word vectorization techniques
    - Different weighting strategies
- You are free to create a new column (or a stream of data) based on the existing columns and use your new column as an independent variable.
- You are welcome to use **any** external dataset to enrich your training and test datasets.
- You are welcome to create **any** logical condition (if, else etc.) to label the target variable (if you do so, please describe why you made these choices).

Please use your real name when you sign up for the Kaggle project. To participate in the competition, please click: https://www.kaggle.com/t/5e8af1fe0d6b4b4b8600c794e8d57ffc.

Model Evaluation

Your submission will be evaluated using the ***Accuracy*** cost function. If you are unsure, please review what ***Accuracy*** means before starting on the project. **Please also report all of your code in the .ipynb file and your confusion matrix both in the .ipynb file and in the report. Please also report *Accuracy* in your code.**

Please use your actual name on Kaggle! [Or, please indicate your nickname in the report!]

Make sure that all of your code is running!

Save the code file you have created as "**kaggle_lab.ipynb**" in the folder you have created at the beginning.

Part III: Creating the lab report (30 points)

Write a report (minimum 2 pages) that includes you name (or your name and your group member's name), all of your findings and the visuals that you created. The report that you will write should use the *IEEE format* and include the following sections:

Abstract: A short summary of your report (This part should include a very brief summary of your methods and analysis and the answer to why you think what you have done is important)

Introduction: A summary of what you expected and did, and two-three of your most significant findings (please use some numerical results here)

Data: Introduce your descriptive findings about the dataset here

Methods: Provide a description of your strategy and the steps you took to improve your prediction model (this includes the steps you followed for data-preprocessing, setting up the model, and checking the strength of the model)

Results: A detailed discussion on the results you obtained. What is your Accuracy value? Evaluate and criticize yourself / your team.

Save the project report as **kaggle_project_report.pdf**.

Final step:

Send your code as an **.ipynb** file and the report in the **.pdf** format through *BlackBoard*.

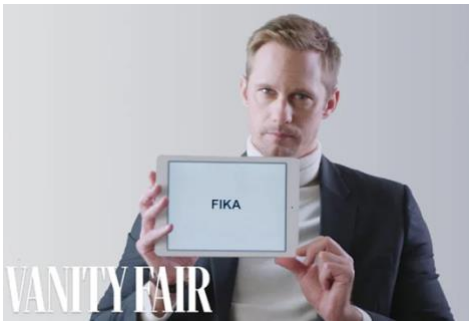
General Rules and Grading

You will be graded based on the following criteria:

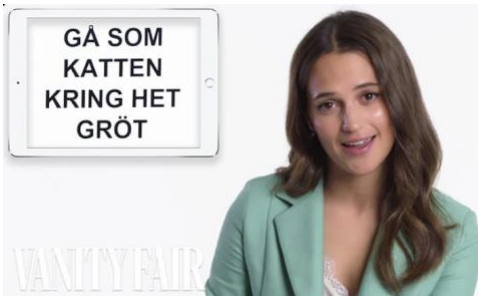
- Code: Cleanliness/understandability (i), executability (ii), format (iii) [We need to be able to run all parts of the code using the datasets provided.]
- Ranking: Ranking in the **Kaggle** competition
- Lab Report:
  - *Introduction* (i), *Data* (ii), *Methods* (iii), *Results* (iv)
  - Flow, readability, level of detail, quality of visuals/tables, adherence to the guidelines

* Around ~5 high-ranked teams will receive a bonus of 0.5% if they choose to present their methods and classification strategy in class.

And, finally, if you are interested in hearing somebody speaking a Northern European language:



https://www.youtube.com/watch?v=KZBNfn3Qo7I



https://www.youtube.com/watch?v=zV_Pwjq7sPc