Name: Vanessa Żyto
Process book

Date of entry: 9.05.2025

What I've worked on: Initial data preprocessing: reading and inspecting the data, cleaning the 'statement' column, removing unwanted characters, lowercasing, punctuation removal

What problems I encountered: What was a bit problematic for me was to think about how 'aggressive' I want to be when it comes to text cleaning; e.g., whether to remove stopwords, preserve numbers, and how much preprocessing might strip valuable information.

What I learned: Practical text cleaning techniques using pandas and regex statements.

Which resources did I use:

- https://docs.python.org/3/library/re.html#regular-expression-syntax
- https://medium.com/@yashj302/text-cleaning-using-regex-python-f1dded1ac5bd
- https://medium.com/@rgr5882/100-days-of-data-science-day-39-removing-unwanted-characters-from-text-columns-5b226a1f2fce
- https://www.geeksforgeeks.org/text-preprocessing-for-nlp-tasks/

Date of entry: 11.05.2025

What I've worked on: Learning about and applying tokenization and lemmatization using NLTK, processing the dataset to generate lemmatized versions of statements, creating customized word clouds for each mental health class to visualize top terms per status.

What problems I encountered:

- unsure whether using stemming or lemmatization; decided to use lemmatization as even though it is slower, it is not likely to produce 'incorrect' words

What I learned:

- explored the difference between lemmatization and stemming
- how to use nltk library, especially nltk.word_tokenizer() and WordNetLemmatizer
- discovered part of speech (POS) tagging
- discovered that lemmatization usually provides cleaner, more interpretable text compared to stemming, but that it's also slower
- how to generate word frequencies and word clouds using matplotlib and WordCloud

Which resources did I use:

- Slides from course ‘Text Retrieval and Mining’ (accessed from a friend who did the course)
- https://www.analyticsvidhya.com/blog/2022/06/stemming-vs-lemmatization-in-nlp-must-know-differences/
- https://www.geeksforgeeks.org/text-preprocessing-for-nlp-tasks/
- https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html
- https://www.youtube.com/watch?v=XnaT_pbEtyI
- https://www.geeksforgeeks.org/python-lemmatization-with-nltk/
- https://www.geeksforgeeks.org/python-lemmatization-approaches-with-examples/

Date of entry: 13.05.2025

What I’ve worked on: TF-IDF → learning of the concept and implementation (first time using it)

What problems I encountered: Choosing optimal parameters for TfidfVectorizer, especially min_df, max_df, and ngram_range. It was a bit hard to estimate at first, so I had to try a lot of different values to discover the effects of it.

What I learned:

- the concept and intuition behind TF-IDF
- how it helps to emphasize important features, and how it is better than e.g., bag of words approach

Which resources did I use:

- https://medium.com/@abhishekjainindore24/tf-idf-in-nlp-term-frequency-inverse-document-frequency-e05b65932f1d
- https://medium.com/@ryblovartem/text-classification-baseline-with-tf-idf-and-logistic-regression-2591fe162f3b
- https://www.youtube.com/watch?v=zLMEnNbdh4Q

Date of entry: 15.05.2025

What I’ve worked on: Exploratory Data Analysis: engineered new features (text_length, number_of_characters), created bar plots, histograms, and explored relationships between text attributes and labels.

What I learned: (a bit more about) visualisation techniques (seaborn)

Date of entry: 16.05.2025

What I've worked on: Introducing and training baseline models such as logistic regression, naive bayes, and Linear SVM and evaluation of them.

What problems I encountered: A bit of problems with choosing the parameters, it was a bit of a trial-and-error approach. Also, at first the sklearn library seemed a bit overwhelming (a lot of features and functions) so I spent quite a lot of time inspecting it.

What I learned:

- Implementation of multiple ML classifiers using sklearn
- Linear SVM algorithm (new for me; didn't learn about it before) → conceptual understanding and sklearn implementation.

Which resources did I use:

- https://scikit-learn.org/stable/modules/svm.html
- https://www.youtube.com/watch?v=_YPScrckx28
- https://www.youtube.com/watch?v=FB5EdxAGxQg

Date of entry: 19.05.2025 10:00 - 13:30

What I worked on: XGB Boost → conceptual understanding & model building and training.

What problems I encountered:

- Very long training time due to large feature set and class imbalance. I managed to reduce the running time by using tree_method='gpu_hist'

What I learned:

- Working with XGB classifier with XBGoost library
- Conceptual understanding of how XGB works
- How boosting differs from other ML classifiers such as logistic regression or SVM

Which resources did I use:

- https://www.youtube.com/watch?v=8b1JEDvenQU
- https://medium.com/@fraidoonomarzai99/xgboost-classification-in-depth-979f11ef4bf9
- https://xgboost.readthedocs.io/en/stable/get_started.html

Date of entry: 20.05.2025

What I've worked on: First attempt at implementing a PyTorch neural network model.

What problems I encountered:

- debugging model structure
- shape mismatches in tensors, converting data into tensors
- running time problems; took some time to run, crashed a lot

What I learned:

- basic PyTorch architecture (model definition, training loop, evaluation)
- solving model shape and data type errors

Which resources did I use:

- https://www.youtube.com/watch?v=tHL5STNJKag
- https://www.youtube.com/watch?v=hrnfKCsSzNY
- https://www.youtube.com/watch?v=E0bwEAWmVEM
- https://medium.com/@sahin.samia/train-a-neural-network-in-pytorch-a-complete-beginners-walkthrough-3897d18d6078

Date of entry: 21.05.2025

What I've worked on: Improved PyTorch model, explored deeper architectures and dropout.

What problems I encountered:

- Limited performance over traditional ML models

What I learned:

- adding hidden layers
- adding dropout
- adding batch normalization

Which resources did I use:

- https://docs.pytorch.org/docs/stable/generated/torch.nn.Dropout.html
- https://www.datacamp.com/tutorial/dropout-regularization-using-pytorch-guide
- https://docs.pytorch.org/docs/stable/generated/torch.nn.BatchNorm1d.html

- https://medium.com/thedeephub/batch-normalization-for-training-neural-networks-328112bda3ae

---

Date of entry: 22.05.2025

What I've worked on: General project cleanup: finalizing markdowns and comments, updating function names, writing code descriptions, improving plot titles and labels.

What problems I encountered: I didn't really write comments before because I thought I'm gonna remember what I did. Well, I didn't.

What I learned:

- the importance of writing comments WHEN you write the code
- Writing cleaner, more readable code
- Importance of commenting and structuring notebooks for reproducibility

---

Date of entry: 25.05.2025

What I've worked on: Final review of notebook and refining evaluations. Cleaning the process book and preparing a project description.