

Final Project

**Online sexism detection**

Course Code: CSE440

| SL | Name | ID |
| --- | --- | --- |
| 1 | Arnab Sarker Sangit | 19201100 |
| 2 | Ahib Afnan Siam | 20201163 |
| 3 | Ramisa Fariha Prova | 20301001 |

# Introduction

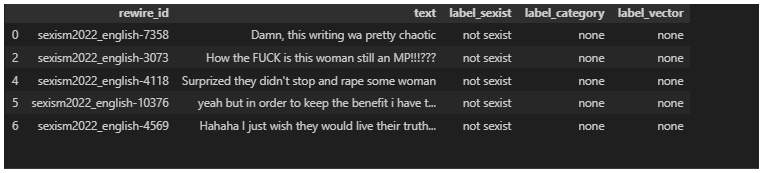
Sexism poses a significant challenge in online environments, impacting women adversely by creating barriers to accessibility and fostering unwelcoming spaces. Moreover, it perpetuates social inequalities and injustices. While automated tools have been developed to detect and evaluate sexist content on a large scale, many only offer broad classifications without detailed explanations. By not only identifying sexist content but also providing insights into why it is considered as such, these tools can enhance interpretability, foster trust, and promote understanding of automated decision-making processes. This approach empowers both users and moderators, fostering a more inclusive and accountable online community.

# Data

The dataset utilized in this project is associated with the SemEval-2023 Task 10: Explainable Detection of Online Sexism, as detailed in the work by Kirk et al. (2023). This dataset, as described on the official competition page authored by Kirk et al., encompasses three distinct classification tasks:

1. Task A: Binary Sexism Detection\*\*: This task involves predicting whether a given post exhibits sexism or not, thus necessitating a binary classification approach.

2. Task B: Category of Sexism\*\*: In this task, posts identified as sexist are further categorized into one of four specific categories: threats, derogation, animosity, or prejudiced discussions. The classification here entails a four-class categorization process.



By leveraging this dataset, the project aims to contribute to the development of automated tools capable of detecting and categorizing online sexism, thereby facilitating a deeper understanding of the phenomenon, and enhancing the interpretability of automated detection systems.

## Online Sexism Detection

In this annotation, the post is annotated in two categories, one is sexist, and one is non-sexist.

## Non-Sexist:

This category represents a post that does not contain any form of prejudice, stereotyping, or discrimination based on sex, and is therefore considered non-sexist.

A blue rectangular bar graph

Description automatically generated

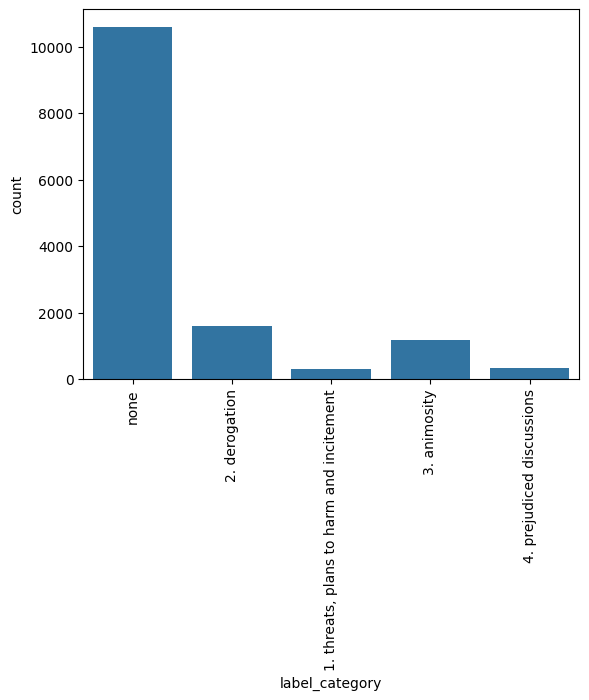
## Sexist

## This category represents a general post characterized by prejudice, stereotyping, or discrimination, typically directed against women solely based on their sex.

## Categorization of offensive language

The provided information outlines the categorization scheme for posts identified as sexist within the dataset. Each category is assigned a numerical label, facilitating classification within the context of the project. Here's a breakdown of the categories and their corresponding numerical labels:

1. Threats, Plans to Harm, and Incitement: Category label 0 signifies posts characterized by expressions of threats, plans to cause harm, or incitement to violence.
2. Derogation: Posts falling under this category are labeled as 1 and typically involve expressions of contempt, belittlement, or disparagement directed towards a particular group, often based on gender.
3. Animosity: Category label 2 denotes posts expressing hostility, animus, or ill-will towards a specific group, primarily based on gender.
4. Prejudiced Discussions: Posts labeled as 3 engage in discussions marked by prejudice, bias, or discriminatory attitudes towards individuals or groups, particularly based on gender.
5. None: Category label 4 indicates posts that do not exhibit any form of sexism or discriminatory content.



A graph with a red line

Description automatically generated

**Task A:**

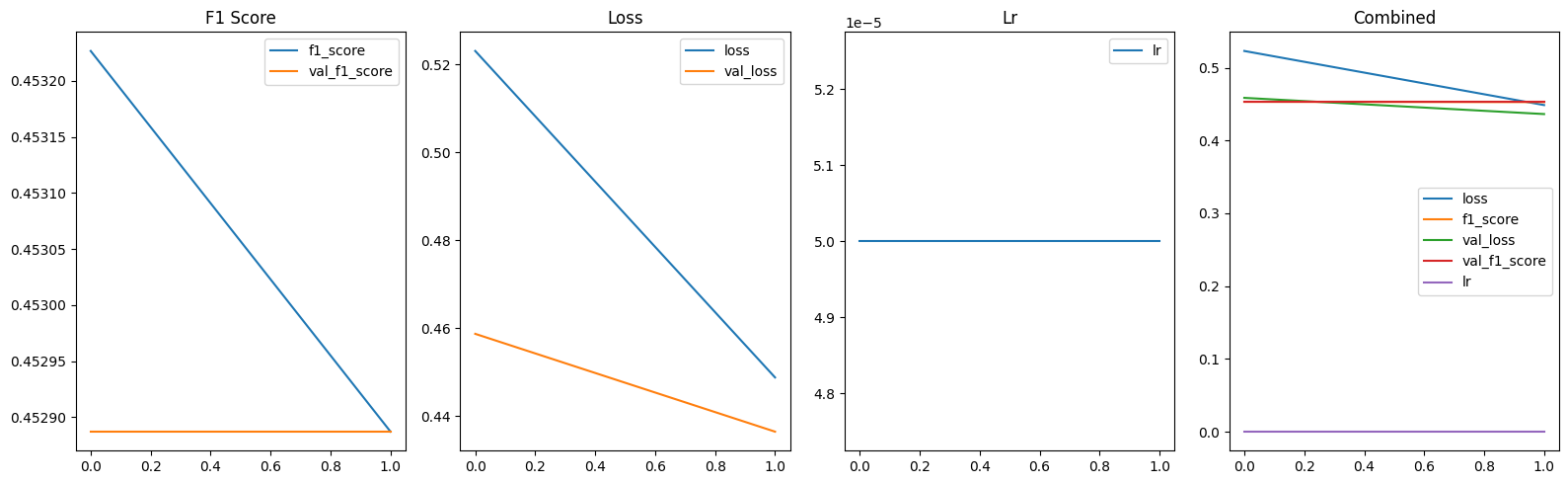
Model: The architecture of the model is meticulously designed for text classification tasks, employing a series of interconnected layers to process input text data and generate classification predictions. The model begins with an input layer, which serves as the entry point for textual inputs. These inputs are then passed through a custom layer, likely implementing a distilled version of the BERT model, to extract contextualized representations of the text. Dropout layers are strategically inserted to mitigate overfitting by randomly deactivating neurons during training. Subsequently, another custom layer, possibly performing sentence-level transformations, processes the input text data further, generating embeddings tailored for the classification task. The model also incorporates multiple dense layers, each with varying numbers of units, to extract higher-level features from the input representations. These dense layers are interconnected, with dropout layers interspersed between them to enhance regularization. Finally, the model's output comprises two sets of dense layers with SoftMax activation, producing classification predictions. These outputs are combined using an "Add" layer to yield the final predictions. With a total of 89,732,868 trainable parameters, this architecture seamlessly integrates BERT-based contextual embeddings with additional layers, aiming to refine representations and enhance the model's capacity for accurate text classification.

A diagram of a flowchart

Description automatically generated

# Results

After running the model, there were the classification report I got



**Task B**

The model architecture is structured to process text data for classification tasks. It begins with an input layer designed to accommodate textual inputs. These inputs are then passed through a custom layer, possibly responsible for converting the text into numerical representations suitable for processing. The subsequent layers consist of convolutional and max-pooling layers, aimed at capturing local patterns and reducing the spatial dimensionality of the feature maps. Following these, flattened and dense layers extract higher-level features and perform classification tasks, with dropout layers incorporated for regularization to prevent overfitting.

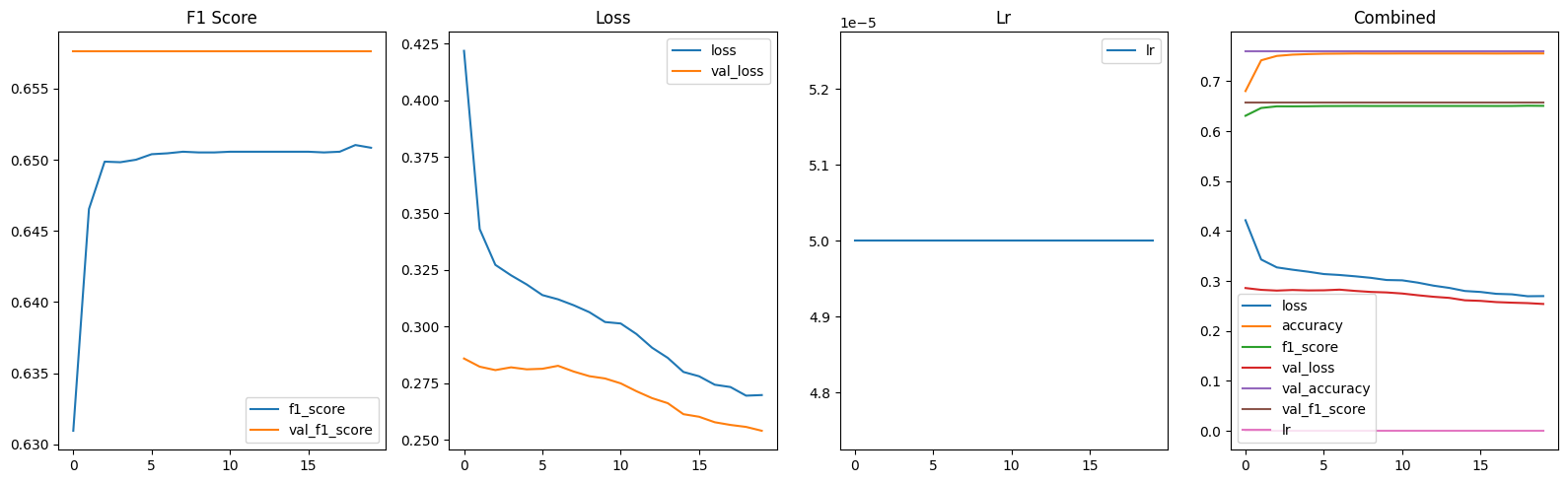
The output layer produces predictions, with a shape corresponding to the number of classes in the classification task, facilitating the classification of input text into predefined categories. Overall, the model comprises trainable parameters totaling 22,939,877, with no non-trainable parameters.

**A diagram of a graph

Description automatically generated with medium confidence**

# Results

After running the model, there were the classification report I got



# Limitation

Every model has some limitations that can vary depending on the type of data. Based on dataset, there are the limitations of used models. Both Model 1 and Model 2 exhibit strengths in text classification tasks but also come with inherent limitations. Model 1's reliance on GPU resources can be a bottleneck, particularly for systems with low computational capabilities or limited GPU specifications, affecting its performance in resource-constrained environments. Additionally, its fixed input size may lead to inefficiencies when handling texts of varying lengths, impacting adaptability across diverse datasets. Despite efficient local feature extraction, Model 1's convolutional filters might struggle to generalize well to unseen data or diverse domains. On the other hand, Model 2's architecture, integrating BERT-based layers, offers sophisticated representations but at a high computational cost, making it challenging to deploy on systems with low-spec GPUs or limited computational resources. The complexity of Model 2's architecture also limits interpretability, hindering understanding of its decision-making processes. Moreover, its dependency on pre-trained

# Conclusion

# In conclusion, the project's exploration of different architectural approaches for text classification highlights the trade-offs inherent in model complexity and computational demands. While the convolutional-based architecture emphasizes efficiency in local feature extraction, it struggles with adaptability to diverse datasets and computational limitations. Conversely, the integration of BERT-based layers offers powerful representations but at the cost of increased computational requirements and reduced interpretability. Going forward, striking a balance between model sophistication and practical deployment considerations will be crucial for developing effective solutions for online sexism detection and similar text classification tasks.

# References

Díaz Redondo, R.P.; Fernández Vilas, A.; Ramos Merino, M.; Valladares Rodríguez, S.M.; Torres Guijarro, S.; Hafez, M.M. Anti-Sexism Alert System: Identification of Sexist Comments on Social Media Using AI Techniques. *Appl. Sci.* **2023**, *13*, 4341. <https://doi.org/10.3390/app13074341>

[HurtBERT: Incorporating Lexical Features with BERT for the Detection of Abusive Language](https://aclanthology.org/2020.alw-1.5) (Koufakou et al., ALW 2020)