Software Engineering for Al-enabled Systems

**Detection Of Online Sexism**

Machine Learning models able to classify the sexist sentences.

#### Tests-Documentation

### Team members:

* Brescia Francesco [f.brescia10@studenti.uniba.it](mailto:f.brescia10@studenti.uniba.it) 789870
* Maria Elenza Zaza [m.zaza16@studenti.uniba.it](mailto:m.zaza16@studenti.uniba.it) 797626
* Grazia Perna [g.perna4@studenti.uniba.it](mailto:g.perna4@studenti.uniba.it) 785321

For more information see [<https://github.com/se4ai2324-uniba/DetectionOfOnlineSexism.git>]

***PREPROCESSING TESTING***

The tests cover scenarios such as punctuation removal, uppercase text transformation, empty input handling, and processing mixed text and numbers. Successful execution confirms the effective preprocessing capability of the clean\_text function.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

***MODEL TRAINING TESTING***

It validates the training accuracy ensuring the model approximates a specified threshold. By confirming conformity to performance expectations, the test contributes to reliable machine learning model development and the assurance of accurate message classification.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

***DATASET TESTING***

This script utilizes the Great Expectations library to perform data validation on three datasets: training, validation, and test sets. The primary goal is to ensure that the datasets meet certain expectations regarding structure and content.

* Model A

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* Model B

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

***BEHAVIORAL DIRECTIONAL TESTING***

This type of test creates messages with distinct tokens and use the predictive models of task A and B to obtain the predicted labels for these messages.

The primary objective is to assert that the predicted labels for messages featuring different tokens are not identical. So this test is designed to verify that the model can distinguish between specific tokens and generate varying predictions based on the token used in the input message.

* Model A

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* Model B

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

***BEHAVIORAL INVARIANCE TESTING***

The objective of this test is to ensure that the model's classification of a message is not influenced by the choice of synonyms like "send" or "give." The test aims to confirm that the model maintains its ability to recognize and categorize messages consistently even when variations in language are introduced.

This evaluation is essential for assessing the robustness of the model's discriminatory language detection across different expressions with similar meanings.

* Model A

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* Model B

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

***BEHAVIORAL MINIMUM FUNCTIONALITY TESTING***

This test serves as a comprehensive test suite for evaluating the performance of the two models of this project. The tests cover various scenarios, including different types of messages and the evaluation of model metrics.

* Model A

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* Model B

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

***TEST API***

Tests concerning APIs implemented with FastAPI are crucial to ensure that the application is robust, reliable, and easy to maintain and develop over time to improve its efficiency. Tests ensure that all the application's functionalities work as expected, verifying that endpoints return the correct data and respond with the appropriate status codes.

When changes are made to the code, tests help ensure that existing functionalities are not accidentally disrupted. Finally, by testing various scenarios, including behavior in abnormal conditions or the submission of invalid input, the application can be ensured to be secure and stable.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente