Cyberbullying detection software documentation

This project features:
Описание проекта:
Download and use
Установка и использование
Statement of Work in Russian (ТЗ ГОСТ-19)
- **1. Введение;**
- **2. Основания для разработки;**
- **3. Назначение разработки;**
- **4. Требования к программе;**
- **5. Требования к программной документации;**
- **6. Технико-экономические показатели;**
- **7. Стадии и этапы разработки;**
- **8. Порядок контроля и приемки.**

Statement of Work in English
ML-model fitting
Обучение модели
Project files
- data
- pages
- .gitignore
- LICENSE
- Presentation.pdf
- README.md
- Datasets_concatenation.ipynb
- Cyberbullying_detection_models.ipynb
- requirements.txt
- run.bat
- streamlit-app.py

This project features:

- A jupiter notebook with all the ML models fitting process (Comments are in Russian).
- A simple streamlit app with EDA step and an access to fitted models.

- A presentation for our project (in Russian).

For model fitting we united two of the following datasets: Cyberbullying Classification and Cyberbullying Dataset.

Created by Karina Zakirova, Vyatcheslav Sviridov, Maxim Shestakov, Anastasia Matushchak, Alexei Bogdan.

View code on GitHub here.

Описание проекта:

- Jupiter notebook файл с тренировкой модели машинного обучения.
- Приложение на основе streamlit с описанием EDA датасета и доступом к обученным моделям.
- Презентация проекта.

Для обучения модели использовались два датасета, объединенные нами в один: Cyberbullying Classification и Cyberbullying Dataset.

Авторы: Karina Zakirova, Vyatcheslav Sviridov, Maxim Shestakov, Anastasia Matushchak, Alexei Bogdan.

Код проекта на GitHub здесь.

Download and use

The program is stored on GitHub here. To download, press the **Code** button and click on **Download ZIP**. Once downloaded, unzip the archive.

One can run the streamlit application with `run.bat` file (on Windows) or using the `streamlit run streamlit-app.py` command from terminal.

Установка и использование

Программа находится в свободном доступе на сайте GitHub по ссылке. Для скачивания нажмите кнопку **Code** и выберите опцию **Download ZIP**. После скачивания распакуйте архив.

Streamlit-приложение может быть запущено путем открытия файла `run.bat` (для Windows) или с помощью использования команды `streamlit run streamlit-app.py` в командной строке.

Statement of Work in Russian (ТЗ ГОСТ-19)

1. Введение;

Основная цель программы “АВТОМАТИЧЕСКАЯ ДЕТЕКЦИЯ КИБЕРБУЛЛИНГА В ТЕКСТОВЫХ СООБЩЕНИЯХ” предназначена для автоматизированного определения негативной информации в текстовых сообщениях. Данная программа может применяться для анализа и модерации комментариев и текстовых сообщений на различных сайтах в сети Интернет. Программа также определяет не только наличие негативной информации, но и тип такой информации (например, религиозная, гендерная или расовая дискриминация). Это позволяет бороться с распространением дискриминации и негативных сообщений в публичном пространстве.

2. Основания для разработки;

На основании приказа преподавателя Факультета Гуманитарных Наук НИУ ВШЭ-НН Кисляниciной Е.А. от 01.12.2023 “О разработке программного обеспечения для детекции буллинга”. Наименование программы: “АВТОМАТИЧЕСКАЯ ДЕТЕКЦИЯ КИБЕРБУЛЛИНГА В ТЕКСТОВЫХ СООБЩЕНИЯХ” (“AUTOMATIC DETECTION OF CYBERBULLYING IN TEXT MESSAGES”).

3. Назначение разработки;

Функциональное назначение: программа предоставляет возможность ввода определенного текста для определения наличия в нем негативной информации. Для негативной информации определяется ее тип. Программа основана на модели машинного обучения, что делает распознавание достаточно точным. Кроме того, программа обладает удобным пользовательским интерфейсом.

Эксплуатационное назначение: программа предоставляет пользователям возможность эффективно определять наличие негативной информации в тексте с целью последующего удаления такой информации. Программа может быть встроена на различные сайты для модерации пользовательской активности.

4. Требования к программе;

Требования к функциональным характеристикам:

- программа принимает на вход текстовые данные, обрабатывает их с помощью модели машинного обучения и выдает сообщение пользователю о наличии или отсутствии признаков кибербуллинга в тексте.

Требования к надежности:

- программа должна обеспечивать высокий уровень точности работы модели машинного обучения.

Условия эксплуатации:

- конечный продукт может быть внедрен на сервер в виде обученной модели машинного обучения для автоматической детекции оскорбительных сообщений в текстах пользователей. Также продукт может быть использован в виде веб-сервиса, который автоматически определяет наличие и тип кибербуллинга в введенном пользователем тексте при помощи обученной модели машинного обучения.

Требования к составу и параметрам технических средств:

- компьютер на Windows, macOS, Linux.

Требования к информационной и программной совместимости:

- программа должна быть выполнена на Python версии 3.8+. Для разработки программы должны быть применены следующие python-библиотеки: pandas, scikit-learn, streamlit, другие библиотеки использовать по мере необходимости.

Требования к маркировке и упаковке:

- программное изделие распространяется по сети Internet. Дистрибуция производится через сайт Github. Специальных требований к маркировке не предъявляется..

Требования к транспортированию и хранению:

- особых требований к транспортировке не предъявляется. Программа хранится в виде архива файлов на сайте GitHub: https://github.com/AcipenserSturio/iad-project

5. Требования к программной документации;

1. “АВТОМАТИЧЕСКАЯ ДЕТЕКЦИЯ КИБЕРБУЛЛИНГА В ТЕКСТОВЫХ СООБЩЕНИЯХ”. Техническое задание (ГОСТ 19.201-78).

6. Технико-экономические показатели;

В рамках данной работы расчёт экономической эффективности не предусмотрен.

Использование программы делает выполнение задачи детекции негативной информации более эффективным, точным и быстрым, что позволит бороться с распространением негативной информации в сети Интернет максимально эффективно.

7. Стадии и этапы разработки;

01.12.2023 — составление ТЗ.
05.12.2023 — готова предобученная модель МО и частично собран датасет.
08.12.2023 — предварительная демонстрация проекта, собран дополнительный датасет.
14.12.2023 — объединение и доразметка датасетов, подготовка финальной презентации проекта.
15.12.2023 — финальная презентация проекта.

8. Порядок контроля и приемки.

15.12.2023 — представление финальной версии проекта.

Statement of Work in English

The main purpose of the program “AUTOMATIC DETECTION OF CYBERBULLYING IN TEXT MESSAGES” is designed to automatically identify negative information in text messages. This program can be used to analyze and moderate texts on various Internet sites. The program determines not only the presence of negative information, but also the type of such information (for example, religious, gender or racial discrimination), thus providing the possibility to combat the spread of discrimination and negative information in the public space.

The program analyses input text and determines the presence of negative information and its type. This piece of software is based on a machine learning model, which makes the recognition quite accurate. In addition, the program has a user-friendly interface.

The given application provides users with the ability to effectively determine the presence of negative information in texts in order to subsequently delete or analyze

such information. The program can be embedded on various websites to moderate user activity.

Functional characteristics:

- The program accepts text data as input, processes it using a machine learning model and sends a message to the user about the presence or absence of signs of cyberbullying in the text.

Reliability requirements:

- The program must ensure a high level of accuracy of the machine learning model.

Operating conditions:

- The final product can be embedded on the server in the form of a trained machine learning model for automatic detection of offensive messages in user texts. The product can also be used as a web service that automatically detects the presence and type of cyberbullying in the text entered by the user using a trained machine learning model.

Software and hardware requirements:

- a computer running on Windows, macOS, Linux;
- Python version 3.8+;
- pandas, scikit-learn, streamlit, and other Python modules.

Distribution:

- The software product is distributed over the Internet. Distribution is done through the Github website. The program is stored as an archive of files on the GitHub website: https://github.com/AcipenserSturio/iad-project

License:

```
MIT License

Copyright (c) 2023 Acipenser Sturio

Permission is hereby granted, free of charge, to any persor
of this software and associated documentation files (the "S
in the Software without restriction, including without limi
to use, copy, modify, merge, publish, distribute, sublicens
```

```
copies of the Software, and to permit persons to whom the S
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY K
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MER
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAG
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERW
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTH
SOFTWARE.
```

ML-model fitting

In order to create a good dataset with enough information we united two datasets in a separate Jupiter Notebook. The resulting dataset had 52538 rows.

Prior to model fitting, the dataset had to be cleared from duplicates and insignificant messages. We noticed the presence of several abnormally large texts. Judging by the lack of contextual connection between its parts, these texts were mistakenly added to the dataset. As a result, we decided to remove such texts from the dataset. To delete large texts, we relied on the pattern rather than the length of the messages, since it is influenced by links added to the message, emoticons, etc., which increase the actual length of the message in characters. We found that the erroneous texts contain a combination of \r\n characters and deleted all texts containing this combination of characters.

Model was split with the use of sklearn.model_selection module and vectorized with TF-IDF vectorizer from sklearn.feature_extraction.text. We tried several classifiers, such as KNN classifier, logistic regression, SVC and LSVC, multinomial naive Bayes, decision tree and Random Forest. Logistic regression model was proved to be the best one. Finetuned model was saved using the pickle module.

Texts were also vectorized with CountVectorizer. The aforementioned classifiers were used as well. LSVC performed better than other models, therefore it was finetuned and saved to a .pkl file. The final average accuracy of the model was 83%.

Обучение модели

Для того, чтобы создать действительно качественную модель, мы решили увеличить количество данных и объединить два датасета в один в отдельном Jupiter Notebookю В результате получился датасет с 52538 строками.

Перед обучением модели набор данных пришлось очистить от дубликатов и незначительных сообщений. Мы заметили наличие нескольких аномально больших текстов. Судя по отсутствию контекстной связи между его частями, эти тексты были ошибочно добавлены в набор данных. В результате мы решили удалить такие тексты. Чтобы удалить большие тексты, мы полагались на шаблон, а не на длину сообщений, поскольку на длину влияют ссылки, добавленные в сообщение, эмотиконы и т.д., которые увеличивают фактическую длину сообщения в символах. Мы обнаружили, что ошибочные тексты имеют комбинацию символов \r\n, и удалили все тексты, содержащие эту комбинацию символов.

Модель была разделена с использованием модуля sklearn.model_selection и векторизована с помощью TF-IDF из sklearn.feature_extraction.text. Мы опробовали несколько классификаторов, таких как KNN classifier, логистическая регрессия, SVC и LSVC, мультиномиальный наивный Байес, Decision Tree и Random Forest. Модель логистической регрессии оказалась лучшей. Доработанная модель была сохранена с помощью модуля pickle.

Тексты также были векторизованы с помощью CountVectorizer. Также использовались вышеупомянутые классификаторы. LSVC работала лучше, чем другие модели, поэтому она был доработана и сохранена в файл .pkl. Итоговая средняя точность модели составила 83%.

Project files

data

This folder contains trained models in .pkl files and the initially used dataframe:

- cvec_lsvc_finetuned_new.pkl contains the SVC model.
- decoder.pkl contains the label decoder.
- tf_idf_logreg_finetuned_new.pkl contains the logistic regression model.
- cyberbullying_tweets.csv contains the dataframe.

Папка содержит уже обученные модели в файлах формата .pkl, а также изначально использованный датасет:

- cvec_lsvc_finetuned_new.pkl содержит SVC-модель.
- decoder.pkl содержит декодер лейблов.
- tf_idf_logreg_finetuned_new.pkl содержит модель логистической регрессии.
- cyberbullying_tweets.csv содержит датасет.

pages

- diagrams.py contains basic EDA and dataframe statistics with plots.
- diagrams.py содержит EDA главную статистику по датасету с графиками.

.gitignore

- .gitignore file specifies intentionally untracked files that Git should ignore.
- .gitignore игнорируемые Git файлы.

LICENSE

MIT Licence file.

Файл с лицензией MIT.

Presentation.pdf

Contains project presentation in Russian language.

Презентация проекта на русском языке.

README.md

README file with project description.

README файл с кратким описанием проекта.

Datasets_concatenation.ipynb

Jupiter Notebook file with dataset concatenation.

Jupiter Notebook для объединения датасетов.

Cyberbullying_detection_models.ipynb

Jupiter Notebook file with model fitting pipeline. EDA, text preprocessing, text vectorization and model fitting and finetuning.

Jupiter Notebook с описанием всего процесса тренировки модели. Содержит EDA, обработку и векторизацию текста, обучение и подбор параметров модели.

requirements.txt

A list of Python modules required to run the program.

Список необходимых для работы программы модулей.

run.bat

File to run the streamlit application on Windows.

Файл для запуска streamlit-приложения на Windows.

streamlit-app.py

Contains the streamlit application that runs the program.

Содержит streamlit-приложение, запускающее программу.