

# Information technology for textual content author's gender and age determination based on machine learning

Victoria Vysotska<sup>1,†</sup>, Lyubomyr Chyrun<sup>2,†</sup>, Sofia Chyrun<sup>1,†</sup> and Mariia Soltys<sup>1,\*</sup>,<sup>†</sup>

<sup>1</sup> Lviv Polytechnic National University, Stepan Bandera 12, 79013 Lviv, Ukraine

<sup>2</sup> Ivan Franko National University of Lviv, University 1, 79000 Lviv, Ukraine

## Abstract

In the process of implementing this project, namely the project on determining the author's age and gender based on his text, a model was developed that determines these biological data of the author based on his text. Before starting work, similar studies on a similar topic are reviewed to find out what has already been researched and tested, and what is still worth investigating. Also, from these studies, it was possible to find many clues about which implementation methods and tools are better to choose, and which work better for this task. The project work is carefully planned using process diagrams and data flows. The best methods and tools for the implementation of this project were studied, and simple classification and regression models of Random Forest became such tools. Such models were chosen, because they cope with the task quite well, and are much less resource-intensive than the same large language models, in addition, they are very easy to use and configure. Two datasets were selected, a dataset with blogs and a dataset with books. The dataset with blogs was used the most because it contains both the age and gender of the blog author. The prediction accuracy of the "book" model is 0.8, and with blogs - 0.6. Before use, the data was analysed and cleaned, later transformed into embeddings and sent for model training. The results of the model are studied and analysed in detail. Many useful features are extracted that are responsible for classifying the age or gender of the author in the texts. In addition, many interesting regularities were observed in the process of analysing the results. Additionally, a test case is implemented that allows the user to easily interact with my model.

## Keywords

machine learning, text analysis, dataset, author, age, gender, NLP, cybersecurity, context, content

## 1. Introduction

The problem of determining the gender and age of the author of the text is a difficult task, especially in the context of the Internet, where information is often provided anonymously


---

MoDaST-2024: 6th International Workshop on Modern Data Science Technologies, May, 31 - June, 1, 2024, Lviv-Shatsk, Ukraine

\* Corresponding author.

† These authors contributed equally.

✉ victoria.a.vysotska@lpnu.ua (V. Vysotska); Lyubomyr.Chyrun@lnu.edu.ua (L. Chyrun); sofia.chyrun.sa.2022@lpnu.ua (S. Chyrun); mariia.soltys.sa.2020@lpnu.ua (M. Soltys)

 0000-0001-6417-3689 (V. Vysotska); 0000-0002-9448-1751 (L. Chyrun); 0000-0002-2829-0164 (S. Chyrun); 0000-0002-5378-4350 (M. Soltys)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

or under pseudonyms [1-5]. Also, this issue is relevant both for the distribution of advertising to the target audience, for example in social networks, and for determining additional parameters of the author of an anonymous text, especially if it is fake/propaganda/disinformation [6-11]. Although there are machine learning models for determining gender and age based on photos or videos, for example, posted on social networks, these approaches have limitations, since real visual information about the author is not always available [12-16]. Considering this, researchers pay attention to text analysis to determine such parameters, which opens up new opportunities [17-22]. Analysis of the text to determine the gender and age of the author depends on various factors, including the style of the author's writing, images, lexical features, and used words and phrases [23-39]. One of the approaches is the application of machine learning methods to textual data [40-45]. For example, models based on neural networks can use the analysis of syntactic and semantic features of the text to determine the gender and age of the author [46-54]. Research in this direction is already underway, and they indicate the potential of these approaches [55-60]. On the other hand, determining gender and age from a text can be a more difficult task due to the variability over time of features of writing styles, context, and other factors [61-69]. Therefore, it remains an active area of research in the field of natural language processing. Finally, the development of new approaches to the analysis of textual information may in the future help to solve the problem of determining the age and gender of the author from his texts on the Internet as an additional parameter for identifying the potential author of the set of generated fakes/propaganda/disinformation.

Determining the age and gender of the author based on the text written by him is a very relevant problem today. Such a model could be useful in various areas, for example, in the field:

- cyber security or law enforcement agencies, to detect and identify persons who plan or commit crimes on the network. It will help in detecting internet fraudsters, and online criminals or even in the investigation of cyber security threats;
- historical research, to determine the authorship of texts or the dating of the writer's works, which can be important for the identification of authors or the analysis of the development of language and styles in different historical periods;
- secondary and higher education to prevent plagiarism and ensure academic integrity. A model for determining the gender and age of the author from the text can help determine whether works written by students or schoolchildren are authentic;
- marketing and analysis of social networks, this model can be useful for determining the target audience, creating personalized offers and analysing user behaviour;
- psychological and sociological research, i.e. it can be useful in psychological and sociological research to understand the peculiarities of language style and psychosocial characteristics of different population groups.

Also, it is worth noting that, in the conditions of war, such a model would be useful for Ukraine to identify collaborators, trolls, propagandists or criminals based on their texts on the Internet and mass media, including in social networks.

The purpose of the research is to develop an information technology for text analysis for features to determine the gender and age of the author based on machine learning.

The object of the research is the process of identifying the linguistic features of the text content to determine the gender and age of the author.

The subject of the research is methods and means of determining the gender and age of authors of texts.

The paper considers the definition of two characteristics at the same time for the first time, which was not previously investigated in other works [61-65]. In addition, this work explores age and gender characteristics reflected in texts, as opposed to identifying these characteristics through images and videos.

## **2. Related works**

In the context of the research area, namely the determination of the gender and age of the author of the text, the need to rely on previous research becomes especially critical. This is due to several factors. Firstly, there are practically no works on this topic in the Ukrainian context, the existing studies were mainly carried out by English-speaking researchers. Secondly, the availability of Ukrainian-language datasets with data on authors (their age and gender) and texts is very limited (if at all), so conducting a study of the Ukrainian language that would not include the creation of a completely new dataset is practically unattainable.

Initial problems with the search for relevant data create difficulties for the implementation of research in the Ukrainian context. Ukrainian data on authors and texts are not available in the available datasets, which makes it difficult to carry out an objective analysis. In this regard, we will focus on English-language studies and datasets to ensure an adequate amount of data for analysis and project development.

This situation highlights the importance of the study taking into account the results of other studies carried out in the English-speaking context and meeting global standards in the field of text analysis to determine the age and gender of the author.

Social media is important for monitoring the perception of public health issues and for educating target audiences about health. However, limited information on the demographics of social media users makes it difficult to identify conversations between target audiences and limits the effectiveness of using social media for public health surveillance and educational interventions [66-75]. Certain social media platforms provide demographic information about the followers of a user's account. If they are provided, they are not always disclosed. Therefore, researchers have developed machine learning algorithms to predict the demographic characteristics of social network users, mainly for Twitter [61]. To date, limited research has been conducted on predicting the demographic characteristics of Reddit users [61]. The study was conducted taking into account data and metadata about Reddit users, that is, not only their posts but also the communities in which they leave their posts, comments or simply subscribe. The researchers manually flagged users' data using the SMART app, looking for confirmation

of their age in comments or posts where users indicated it themselves. Data volumes were such that each age category (youth (13-17 years), young adults (18-20 years), and adults (21-54 years)) had a minimum of 625 records. Metadata was collected after tagging the data by age, via the Reddit API for each user. Metadata included user-level information (e.g., year of account creation), submission-level data (e.g., post popularity), and comment-level data (e.g., commenting frequency). The study focused on specific metadata that could potentially help distinguish between adolescent and adult age groups. The research identified 1,523 variables that could potentially indicate the age of Reddit users:

- Final statistics: average level of evaluation of publications, etc.
- Frequency of subreddits: frequency of posts in specific subreddits related to age groups.
- Frequency of emoji usage: Frequency of emoji usage in comments.
- Post Patterns: Percentage of posts that were videos, images, etc.
- Use of terms: TF-IDF scores for specific terms (e.g. "school") used in comments.

The dataset is divided into train and test (80/20), after which various models (logistic regression, random forest, k-nearest neighbours, Gradient boosted trees) that could potentially show a good result for this task were collected and evaluated by their indicators such metrics as AUROC, precision, recall and F1 score. The best result was shown by the Gradient boosted trees model (F1 score: 0.77, AUROC: 0.84). In the end, it is analysed and evaluated which of the signs have the greatest influence on determining the age of users. This study is important because it helps to better understand what should be relied on when determining the age and gender of the authors of the texts, and which signs are the most important and influential.

The rapid growth of social networks has generated an unprecedented amount of user-generated data, which provides an excellent opportunity for text mining [62]. The main purpose of authorship analysis, an important part of text analysis, is to learn as much information as possible about the author of the text through the subtle variations in writing styles that exist within genders, ages, and social groups. Such information has a variety of uses, including advertising and law enforcement. One of the most accessible sources of user-generated data is Twitter, which provides free access to most user data through its Data Access API. In the study [62], the authors sought to determine the gender of Twitter users using Perceptron and Naïve Bayes with selected parameters from 1 to 5-gram features from the tweet text. Stream applications of these algorithms have been used for gender prediction to process the speed and volume of tweet traffic. Since informal text such as tweets cannot be easily evaluated using traditional dictionary methods, the study [62] implemented n-gram features to represent streaming tweets. The large number of 1- to 5-grams requires only a subset of them to be used in gender classification, for this reason, the informative features of n-grams are selected using several selection algorithms. In the best case, the Naïve Bayes and Perceptron algorithms showed accuracy, balanced accuracy and F-measure above 99%.

The study [62] is based on the analysis of messages and posts on Twitter, and the main goal of the study is to extract signs that would indicate some personal information

about the author of the tweet. The peculiarities of this study are that informal language is used in twitter, and this paper is devoted to the actual analysis of informal language for important identification features. This approach has its difficulties, because, first of all, Twitter has a limit of 140 characters per message, which is a problem for traditional text analysis, as large segments of texts are usually used in such analysis. Secondly, since it is an informal language, users very often use acronyms, so-called text emoticons, and especially distorted spelling of the word, which can also make analysis more difficult. Before conducting the study, the data was carefully filtered and manually labelled using the API. Six different feature selection mechanisms were used to identify them and determine which ones would best help accomplish the task. This process aims to extract the most informative n-grams from tweets to improve gender prediction accuracy. To perform the task of classification, a simple neural network, namely Naïve Bayes, is used, which is based on Bayes' theorem. The importance of the study [62] is that it nicely highlights the difficulties in analysing spoken language and informal writing. Like the previous one, this study also highlights the importance of the correct choice of features to improve the accuracy of the model's prediction and, accordingly, the accuracy of the author's gender classification.

In [63], it was investigated whether wording, stylistic choices and online behaviour can be used to predict the age category of blog authors. The authors hypothesize that significant changes in writing style distinguish pre-social media bloggers from post-social media bloggers. By experimenting with different years, the authors found that college students' birth dates around the time when social networking sites like AIM, SMS texting, MySpace, and Facebook became popular gave accurate age predictions. The authors also determined that the characteristics of Internet writing are important characteristics for predicting age, but lexical content is also necessary to obtain significantly more accurate results. Our best results provide an accuracy of 81.57%.

The basis of this study [63] is the determination of the age of blog authors. The definition is based on stylistic choices and online behaviour. The best part of the model is to determine the approximate age of a person, namely, whether he was born before the era of social networks, or already during it. The blogs are collected from the LiveJournal magazine, namely those blogs where the age of the author is indicated. All the articles are from American bloggers.

Several features have been identified that help determines the author's age, including special words, stylistic features such as slang or text emoticons, as well as online behaviour such as frequency of posting and number of friends. A binary classification model based on year of birth was used, slightly modified to address changes in blogging styles based on popular social media technologies. In a study [63], it was found that two age groups (born in 1977-1979 and born in 1982-1984) differed greatly in terms of blogging style. Both stylistic and substantive features strongly influenced the prediction of age with the help of other variables that helped in determining the age group. The study [63] is important, because it notes the determination of age purely by text analysis and the use of certain features in the text, without taking into account metadata about the user. The research can be expanded to determine the geographical location or other data about the author.

Although the study of the relationship between discourse patterns and personal identity has been going on for decades, the study of these patterns using language technologies is relatively recent [64]. In this latest tradition, the authors in [64] implemented the prediction of the author's age from the text as a regression problem. They investigated the same task using three very different genres of data simultaneously: blogs, telephone conversations, and online forum posts. A domain adaptation technique was also used, which allows for training a joint model including all three corpora together as well as separately and analysing the differences in predictive performance between the combined and corpus-specific aspects of the model. Effective features include both stylistic (such as POS templates) and content-oriented features. Using a linear regression model based on shallow text elements, the authors in [64] obtained correlations up to 0.74 and mean absolute errors between 4.1 and 6.8 years. In the study, three datasets were selected for analysis: blog corpus, fisher telephone corpus, and breast cancer forum. Each dataset has a different age distribution, which affects the determination of the age of users. The blog dataset has more young people, while the breast cancer forum dataset has more older people. The telephone conversation dataset has the most balanced age distribution. There were four different linear regression models for predicting user age. Interestingly, the study [64] states that the gender of the user significantly affects the identification of his age, that is, it makes sense to determine both characteristics. The best results were obtained by the dataset of telephone conversations, immediately followed by the dataset with blogs. The study also provides examples where the signs that can be used to determine the age of an Internet user are visible.

There is a growing interest in automatically predicting the gender and age of authors based on texts. However, most research so far ignores that language use is related to the social identity of speakers, which may differ from their biological identity. In [65], the authors combined insights from sociolinguistics with data collected through an online game to highlight the importance of approaching age and gender as social variables rather than static biological variables. In the study, thousands of players guessed the gender and age of Twitter users based on tweets alone. The authors showed that more than 10% of Twitter users do not use language that the crowd associates with their biological sex. It has also been shown that older Twitter users are often perceived as younger than they are. The authors' conclusions highlight the limitations of current approaches to gender and age prediction from texts. This is quite an interesting study that calls into question all previous studies. The authors point out that often the behaviour of users does not correspond to their biological age or sex, so it makes sense to define gender as a social construct, and not as a biological feature, the same applies to age. It's common for people on Twitter to post messages that don't match their gender or age. The research was conducted using a game developed by the authors, where people guessed the gender and age of a certain author from Twitter. Thousands of participants joined the game and the result showed a significant difference in the guessed age and the real age of the authors, using only the text of the tweets. According to a study [65], 10% of Twitter users and their language are not associated with their real age or gender. Also, older Twitter users are often classified as younger. With this study, the authors highlighted the problem that the automatic determination of age or gender is often based

on stereotypical features, which in reality may not correspond to reality at all. This limits the models in their ability to draw on upbringing and social constructs rather than just biological age. The authors of the study call for consideration of social and sociocultural influence and the variability of people's pronunciation when developing classification models.

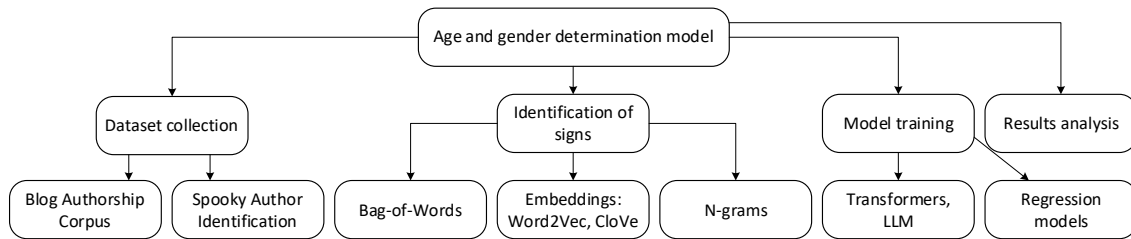
### 3. Methods and materials

Many studies highlight the main characteristics by which it is possible to identify age or gender, which we could use in our study [61-75]. Different studies have used different data and different models to predict user characteristics [61-75]. This allows you to compare them and understand what could be used in your research. For example, the study [64] analysed how the different distribution of data in the dataset affects the accuracy of the model and, accordingly, the accuracy of the characteristics predicted by it, i.e. age or gender. Research [65] allows us to look at our topic from a critical point of view, and to determine what should be taken into account when developing one's program, namely, the fact that the author's behaviour may often not coincide with his biological sex or age due to certain social constructs or upbringing. To do this, we will first define the tree of the goals of our research.

A tree of goals is a hierarchical tree-like structure obtained by dividing the overall goal into subgoals, which in turn can also be divided into smaller subgoals, functions, etc. (Fig. 1). Graphically, the tree is depicted with "branches down", and the main goal is placed at the highest level. The advantage of building a goal tree is the possibility of dividing a large unfathomable goal into simpler tasks that can be solved by known methods. At the root of the tree is "Development of a model for determining the age and gender of the authors of the text", and the branches of the tree go down from the root:

- Collection of datasets: preparation of datasets for model training and task execution.
  - a. Blog Authorship Corpus - a dataset with blogs and information about the author to determine age and gender [73].
  - b. Spooky Author Identification - a dataset with famous authors and excerpts from their works, for determining gender.
- Feature extraction: selection and ranking of the best features that best influence the model output.
  - a. Bag-of-Words - uses TF-IDF technology.
  - b. N-grams - includes sequences of word combinations (bigrams, trigrams) as features to capture the context.
  - c. Embeddings, Word2Vec, GloVe - turns words into dense vectors that capture semantic meaning.
- Model training: training of the selected model on cleaned data.
  - a. Transformer model - already trained large language models, suitable for gender determination.
  - b. Regression model - models working based on a regression function are suitable for determining age.

- Analysis of results: construction of graphs, statistical analysis, summarization of conclusions.



**Figure 1:** Tree of goals

The methodology of functional modelling is used to create a functional model that reflects the structure and functions of the system, as well as the flows of information and material objects connecting these functions. The IDEF0 diagram was designed to display mechanisms and instructions in the diagram (Fig. 2-3). The main process is to create a model for determining the age and gender of the author based on their text.

*Input:* Excerpts or fragments of texts by different authors.

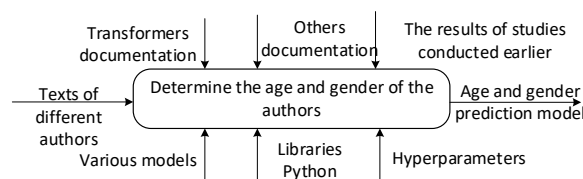
*Output:* Age and gender prediction model.

*Mechanisms:*

- A wide selection of models can be applied for this task.
- Python libraries allow you to perform a variety of tasks, from pre-processing to data analysis.
- Hyperparameters that can be adjusted to get the best results.

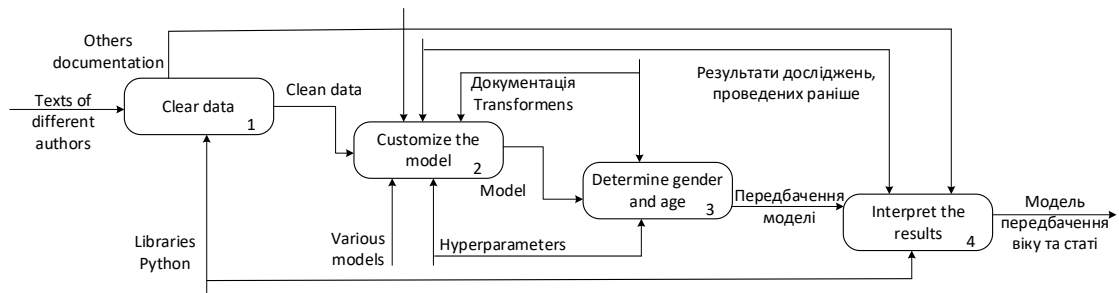
*Instructions:*

- Transformers documentation for proper use of large language models.
- Previous research from which useful information can be gleaned for my research.
- Other documentation will help in the use of numerous libraries in the process of working and developing the model.



**Figure 2:** IDEF0





**Figure 3:** Decomposed IDEF0

A Data Flow Diagram or DFD is a graphical structural analysis methodology that describes external to the system data sources and destinations, logical functions, data flows and data stores that are accessed (Fig. 4-5). That is, the data flows implemented in the project are described.

*Data repositories:*

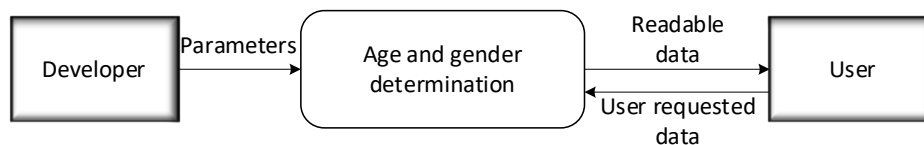
- Blog dataset - downloaded Blog Authorship Corpus dataset [73].
- Book dataset - downloaded Spooky Author Identification dataset [74].
- Documentation - all documentation that controls the developed models and software part of the project.

*External entities:*

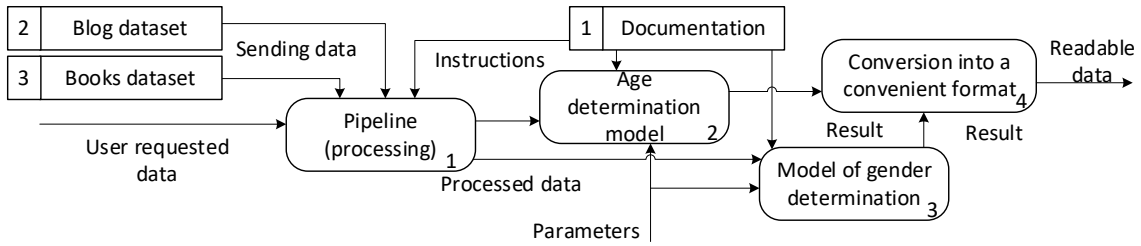
- Developer - a person who develops a model, and configures it.
- User - a natural person who uses a ready-made model.

*Functions:*

- Pipeline - the process of pre-processing data, and preparing them for use by the model.
- Age determination model - a machine learning model that predicts the age of the author based on the texts written by him.
- Gender determination model - a machine learning model that predicts the gender of the author based on the texts written by him.
- Conversion into a convenient format - conversion of the information provided by the model into a convenient and human-readable format using graphs and conversion functions.

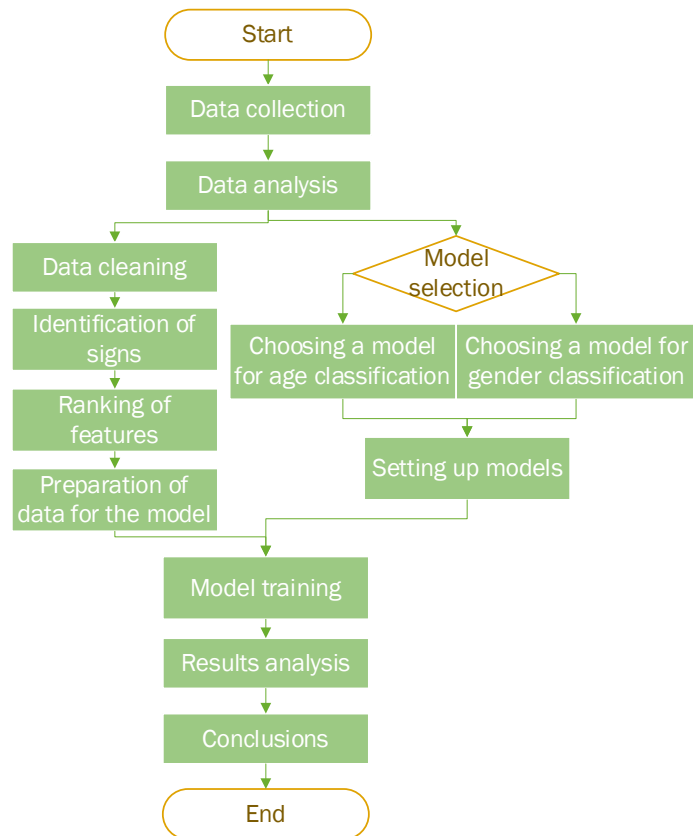


**Figure 4:** Data Flow Diagram



**Figure 5:** Decomposed data flow diagram

A workflow diagram (process diagram) is used to model the sequence of steps or stages in the work process. The main purpose of such a diagram is to visualize and analyse the workflow to optimize or automate the process. For the project, this is a visualization of the development process and all its stages (Fig. 6). In the end, a fully functional model was obtained for determining the age and gender of the author of the text.



**Figure 6:** Workflow diagram

- Collection of data, i.e. datasets with data on authors of texts [73-74].
- Data analysis, identification of data types, their quantity and other metadata for model selection.

The division into branches. Left branch:

- Data cleaning, removal of special characters, unnecessary characters, and articles.
- Identification of features using the previously described methods.
- Ranking of features to determine the most important for this study.
- Preparation of data for sending to the model.

Right branch:

- Selection of models:
  - a. A model for classifying authors by age.
  - b. A model for classifying authors by article.
- Setting up the model, selecting parameters, optimizers and modifying the architecture.

Joining branches:

- Training of the previously configured model on prepared data.
- Evaluation of results using metrics, graphs and analytics.
- Formation of research conclusions.

#### **4. Statement and justification of the problem**

*Statement of the problem:* this study allows us to study the problem of determining the gender and age of the author based on the texts written by him. Its essence is to create a machine learning model to analyse the text and determine the biological data (age and gender) of its author based on the sample of his text.

*Technical characteristics:* as an input, the model accepts a text sample in text format (string, char), processed and cleaned, and as an output, the age, numerical value or numerical interval, as well as gender, and binary value will be analysed.

*Business processes:*

*Data collection → Data processing → Model selection → Model training → And creating a practical application for the model, for example in cyber security for identification.*

*Technical means of implementation:*

- Bag-of-Words, N-grams, Word2Vec, and GloVe are used for data processing.
- To build a model: Transformers, Tensorflow, Keras, PyTorch.

*Application:* the model is developed for research purposes to expand the issue of determining the gender or age of the authors of texts, but it can also be used to identify a person or verify authorship.

*Expected effects:* contributing to research on the identification of biological data of the author from his text. Development of a potentially useful model in cyber security and

other fields. Gaining new knowledge about the development of language models and conducting research.

## **5. Comparison of methods and means of the product under development**

### **5.1. Machine learning models**

Regression models are better for predicting age, here are a few basic ones in comparison:

1. Linear regression. Pluses:

- Simple and clear.
- Fast learning and getting results.

Cons: Assumes a linear relationship between traits and age, which may not be true for complex textual data.

2. Support Vector Regression (SVR). Pluses:

- Effective in large multidimensional spaces.
- Can capture complex relationships using kernel features.

Cons: Requires careful tuning of hyperparameters.

3. Gradient Boosting Regression (for example, XGBoost). Pluses:

- Resistant to fuzzy and noisy data.
- Can effectively capture non-linear relationships.

Cons: Higher computational cost compared to linear models.

Options for using large language models (LLM) to accomplish this task are also considered:

4. BERT (Transformer Bidirectional Encoder Representation). Pluses:

- Captures the bidirectional context in the text.
- Can handle complex relationships and semantics in textual data.
- Pre-trained on a large corpus (e.g. Wikipedia, books) and then customized for specific tasks.

Cons:

- Requires significant computing resources for training and results.
- A large amount of memory.

5. GPT (generative pre-trained transformer). Pluses:

- Creates coherent text appropriate to the context.
- Useful for creating text predictions.

Cons: Can't directly output predicted age or gender; requires additional fine-tuning for a specific task.

## 5.2. Comparison factors

1. Productivity. LLMs are generally excellent at capturing complex patterns and semantics in textual data, potentially leading to higher predictive accuracy compared to traditional regression models.
2. Interpretability. Traditional regression models, such as linear regression, offer straightforward interpretation, making it easier to understand the relationship between characteristics and predictors. LLMs, being deep learning models, are more complex and less interpretive, although techniques such as attention mechanisms can provide some insight.
3. Resource requirements. LLMs require significant computational resources (e.g., GPU, memory) for training and inference due to their deep architecture and large parameter size. Traditional regression models are smaller in terms of resource requirements.
4. Possibility of adaptation to specific tasks. LLM can be customized for specific tasks, such as age and gender prediction, using transfer learning using pre-trained models. Traditional regression models may require more complex work with features and additional tuning for a specific area. Therefore, both regression models and LLM models are suitable for the task of this study. They show different performances for different tasks, so it's best to use several models in your work, give them different parts of the task and compare their performance.

## 6. Experiments

The basis of the project, which fulfils its main goals, namely the determination of age and gender, is a machine learning model written in the Python programming language. Despite this, the model itself takes up relatively little space in the program, and most of it is occupied by data processing and analysis, and analysis of results. It is useful to consider all these parts of the program separately to be able to focus on the methods and processes of each stage.

### 6.1. Data analysis

At the stage of data analysis, the dataset itself is loaded [73-74], and its content, amount of data, data distribution, and search for correlation between data using graphs and other tools are analysed.

- Methods: data loading, manual data cleaning, data visualization.

- Tools: Python libraries (pandas, matplotlib, seaborn).
- Process description: First, the data (dataset) is loaded into the Python environment for further processing. There is a manual review of the dataset and the selection of suitable features in the data. Unnecessary features can be deleted. Next, we build several visualizations using Python libraries to better capture data correlation and create an idea of how to work with them, namely a graph of gender distribution and age distribution in the dataset to check its weighting.

```
df = pd.read_csv('./blogtext.csv')
df.head()
✓ 8.1s
```

	id	gender	age	topic	sign	date	text
0	2059027	male	15	Student	Leo	14,May,2004	Info has been found (+/- 100 pages,...
1	2059027	male	15	Student	Leo	13,May,2004	These are the team members: Drewe...
2	2059027	male	15	Student	Leo	12,May,2004	In het kader van kernfusie op aarde...
3	2059027	male	15	Student	Leo	12,May,2004	testing!!! testing!!!
4	3581210	male	33	InvestmentBanking	Aquarius	11,June,2004	Thanks to Yahoo!'s Toolbar I can ...

```
df = df.drop(columns=['id', 'date', 'sign']) #deleting unnessesary columns
df
✓ 0.0s
```

	gender	age	topic	text
0	male	15	Student	Info has been found (+/- 100 pages,...
1	male	15	Student	These are the team members: Drewe...
2	male	15	Student	In het kader van kernfusie op aarde...
3	male	15	Student	testing!!! testing!!!
4	male	33	InvestmentBanking	Thanks to Yahoo!'s Toolbar I can ...
...	...	...	...	...
681279	male	23	Student	Dear Susan, I could write some really ...
681280	male	23	Student	Dear Susan, I have the second yeast l...
681281	male	23	Student	Dear Susan, Your 'boyfriend' is fuckin...
681282	male	23	Student	Dear Susan: Just to clarify, I am as...

Figure 7: Manual data cleaning

## 6.2. Data processing (pre-processing)

In the data pre-processing stage, the dataset goes through detailed processing and text cleaning to clean the text of unnecessary characters that can negatively affect the accuracy of the model's predictions, as well as converting the data into a numerical format that the model understands and can work with.

- Methods: removal of unnecessary symbols, removal of stop words, tokenization of sentences, lemmatization of words, division of data into sets, vectorization of words, labelling of evaluations.
- Tools: Python libraries (pandas, NLTK).
- Process description: The data from the previous step is first separated into text and scores. Scores are converted into binary (gender) and categorical (age) or numeric (age) formats. After that, the text data is cleaned. First, all uppercase letters are converted to lowercase, sentences are cleaned of stop-words, all kinds

of signs and markings, and, if necessary, lemmatized (in my case, this step turned out to be unnecessary). In the end, already cleaned data are divided into training and test sets.

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')

def stopwords_removal(text):
    new_words = word_tokenize(text)
    new_filtered_words = [
        lemmatizer.lemmatize(word.lower()) for word in new_words if word.lower() not in stopwords.words('english')
    ]
    return ' '.join(new_filtered_words)

✓ 0.7s
outputs are collapsed ...

df = df.sample(n=len(df))
df_short = df[:20000]

✓ 0.1s
```

**Figure 8:** Text processing as data pre-processing (removal of stop words and lemmatization)

```
def text_preprocess(text): #10 minutes
    text = re.sub(r'<.*?>', '', text)
    text = re.sub(r'\W+', ' ', text)
    text = text.lower()
    text = re.sub(r'\n', '\n', text)
    text = re.sub(r'\urlink', '', text)
    text = re.sub(r'im', 'i am', text)

    return text

df_short['clean_text'] = df_short['text'].apply(text_preprocess)
df_short['clean_topic'] = df_short['topic'].apply(text_preprocess)

df_short['clean_topic'] = df_short['clean_topic'].apply(stopwords_removal)
df_short['clean_text'] = df_short['clean_text'].apply(stopwords_removal)

df_short['combined_text'] = df_short['clean_text'] + ' ' + df_short['clean_topic']

df_short['gender_bi'] = df_short['gender'].map({'male': 1, 'female': 0})

x_train, x_test, y_train, y_test = train_test_split(df_short['combined_text'], df_short[['age', 'gender_bi']], test_size=0.2, random_state=42)

✓ 8m 28.0s
```

**Figure 9:** Clear text

### 6.3. Model training

After cleaning and pre-processing the data, it can be transformed into a set of vectors and fed into a model to make predictions. The model itself consists of two Random Forest models, one of which allows classifying age and the other gender. The prediction accuracy of both models is evaluated using metrics.

- Methods: text vectorization, model training, model evaluation.
- Tools: Python libraries (scikit-learn).

- Process description: the data completely cleaned at the previous stage is transferred to the vectorization function, which converts tokens into digital values (embeddings). In this, numerical, form, the data can be transferred to the model for training. Random Forest Classification models were used to classify age and sex, and a Random Forest Regressor was used to determine the numerical value of age. The text and the mark to it are transferred to the model, thus the process of training the model takes place. Next, the model is evaluated and its accuracy is determined by comparing its predictions with real marks.

```
vectorizer = TfidfVectorizer(max_features=10000, ngram_range=(1, 2))
x_train_tfidf = vectorizer.fit_transform(x_train)
x_test_tfidf = vectorizer.transform(x_test)

from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.linear_model import LinearRegression, Ridge, Lasso

# Train a Random Forest classifier for age prediction
rf_age_category = RandomForestClassifier(n_estimators=100, random_state=42)
rf_age_category.fit(x_train_tfidf, y_train['age_category'])

age_category_score = rf_age_category.score(x_test_tfidf, y_test['age_category'])
print(f'Random Forest age category prediction accuracy: {age_category_score}')

Random Forest age category prediction accuracy: 0.64325

age_category_predictions = rf_age_category.predict(x_test_tfidf)
print(classification_report(y_test['age_category'], age_category_predictions))

importances_age_category = rf_age_category.feature_importances_
indices_age_category = np.argsort(importances_age_category)[::-1]
```

Figure 10: Age model training

```
rf_gender = RandomForestClassifier(n_estimators=100, random_state=42)
rf_gender.fit(x_train_tfidf, y_train['gender_bi'])
rf_gender.score(x_test_tfidf, y_test['gender_bi'])

gender_predictions = rf_gender.predict(x_test_tfidf)
print(classification_report(y_test['gender_bi'], gender_predictions))

precision    recall  f1-score   support

0           0.63     0.73     0.68     2019
1           0.67     0.56     0.61     1981

accuracy          0.65     4000
macro avg         0.65     0.65     0.65     4000
weighted avg      0.65     0.65     0.65     4000

importances_gender = rf_gender.feature_importances_
indices_gender = np.argsort(importances_gender)[::-1]

top_features_gender = [vectorizer.get_feature_names_out()[i] for i in indices_gender[:top_n]]
print(f'Top {top_n} features for gender prediction: {top_features_gender}')

Top 15 features for gender prediction: ['technology', 'indank', 'student', 'love', 'internet', 'know', 'go', 'like', 'get', 'really', 'arts', 'one', 'hi', 'think', 'day']
```

Figure 11: Sex model training

## 6.4. Evaluation of results

- Evaluation of the results is almost the most important stage of any research. It allows you to see certain regularities between the results and the initial data, which can sometimes even initiate another study. Data visualization, model



accuracy measurement, feature selection, comparison of predictions with real results, and other methods are used to evaluate research results.

- Methods: visualization of results, construction of predictions, transformation of predictions into a human-understandable format, comparison of data, calculation of numerical metrics.
- Tools: Python libraries (matplotlib, seaborn, scikit-learn, pandas, NumPy).
- Process description: the model predicts age and gender on test data, compares its results with the real ones, and generates graphs. In the work, the most influential signs, by which the model determines age and gender, were identified, and they were displayed in the form of a graph (separately for age and gender). These graphs are among the most important because they help us understand which words can indicate the biological data of the author of the text. In addition, many graphs are created that describe the accuracy of the model, these include the ROC curve, the positive/negative true/false matrix, the histogram of true and predicted age (for numerical age prediction), the distribution of true and predicted age categories (for categorical age determination).

## 6.5. For the user

This stage is created for interaction with the user, it provides an opportunity to enter your text excerpt to determine the gender and age of the author, and the results are presented as clearly as possible for users.

- Methods: calling previously developed functions, outputting results in a human-understandable format.
- Tools: Tools: Python libraries (scikit-learn, NLTK).
- Process Description: The task of this stage is to create an extremely simple and concise section for user interaction. The user's task is to enter the text in the right place, the author of which needs to be determined, and run two cells with the code. The text entered by the user is passed to previously developed functions, undergoes cleaning, removal of stop-words, transformation, vectorization, transfer of text to the model, conversion of the text into a readable format and output of the results to the user. The whole process takes 8 lines and takes no longer than a minute.

```
your_text = "I would love to go to the gallery with you! I heard they are exhibiting the most famous paintings of Monet, he is my favourite painter, I am so excited!"

#now run this cell and it will give you result

text = text_preprocess(your_text)
text = stopwords_removal(text)

text = pd.Series(your_text)

text = vectorizer.transform(text)

pred_age = rf_age_category.predict(text)
pred_gen = rf_gender.predict(text)

result = decategorize(pred_age, pred_gen)
print(f'The author\'s age is in this range: {result[0]}\nAuthor\'s gender: {result[1]}')

The author's age is in this range: ['20-30']
Author's gender: ['female']
```

Figure 12: For the user

## 6.6. User manual

To run the program, the user's device must meet the following requirements:

- Internet connection;
- Operating system: Windows 7 or higher;
- Software: a program that supports the .ipynb format (Jupyter notebook, Google Colab web resource, VS Code);
- Features: 8+ GB RAM, CPU (or use Google Colab).

To use the program, you need to follow the following steps:

1. Place the program file and the dataset in one folder.
2. Open the program file.
3. Run each cell individually, one by one, using the start button (usually a trident) to the left of each cell, or the "Run All" button on the top panel of the program, if there is one.
4. Wait until the end of execution of all cells (approximately 10-15 minutes).
5. In the "For user" section (at the end of the file), you can enter the text to be classified, then run the cell with the entered text and the following text, the result will appear under the second cell.

## 6.7. Program code

Downloading required libraries:

```
import numpy as np
import pandas as pd
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
```

Loading dataset:

```
df = pd.read_csv('./blogtext.csv')
df.head()
```

Deleting unnecessary lines:

```
df = df.drop(columns=['id', 'date', 'sign']) #deleting unnecessary columns
df
```

The function of the graph of the distribution of data by gender:

```
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='gender', color='purple')
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```

The function of the graph of the distribution of data by age:

```
plt.figure(figsize=(10, 6))
sns.histplot(data=df, x='age', bins=20, kde=True, color='purple')
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Count')
```

```
plt.show()
```

Data cleaning function from stop-words:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
# nltk.download('punkt') # nltk.download('stopwords')
# nltk.download('wordnet') # nltk.download('omw-1.4')
def stopwords_removal(text):
    new_words = word_tokenize(text)
    new_filtered_words = [#lemmatizer.lemmatize(word.lower()) for word in new_words if
word.lower() not in stopwords.words('english')]
    word for word in new_words if word.lower() not in stopwords.words('english')]
    return ' '.join(new_filtered_words)
```

Sampling part of the dataset (30,000 samples):

```
df = df.sample(n=len(df))
df_short = df[:30000]
```

Converting age into categories (optional):

```
def categorize_age(age):
    if age < 20:
        return 0 # less than 20
    elif 20 <= age <= 30:
        return 1 # 20-30
    else:
        return 2 # more than 30
df_short['age_category'] = df_short['age'].apply(categorize_age)
```

The function of removing unnecessary characters and applying all preprocessing functions to the text:

```
def text_preprocess(text): #10 minutes
    text = re.sub(r'<.*?>', '', text)
    text = re.sub(r'\W+', ' ', text)
    text = text.lower()
    text = re.sub(r'\n', ' ', text)
    text = re.sub(r'urllink', '', text)
    text = re.sub(r'im', 'i am', text)
    return text
df_short['clean_text'] = df_short['text'].apply(text_preprocess)
df_short['clean_topic'] = df_short['topic'].apply(text_preprocess)
df_short['clean_text'] = df_short['clean_text'].apply(stopwords_removal)
df_short['clean_text'] = df_short['clean_text'].apply(stopwords_removal)
df_short['combined_text'] = df_short['clean_text'] + ' ' + df_short['clean_topic']
df_short['gender_bi'] = df_short['gender'].map({'male': 1, 'female': 0})
x_train, x_test, y_train, y_test = train_test_split(df_short['combined_text'],
df_short[['age_category', 'gender_bi']], test_size=0.2, random_state=42)
```

Vectorization of text data:

```
vectorizer = TfidfVectorizer(max_features=10000, ngram_range=(1, 2))
x_train_tfidf = vectorizer.fit_transform(x_train)
x_test_tfidf = vectorizer.transform(x_test)
```

A model for determining age:

```
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.linear_model import LinearRegression, Ridge, Lasso
rf_age_category = RandomForestClassifier(n_estimators=100, random_state=42)
rf_age_category.fit(x_train_tfidf, y_train['age_category'])
age_category_score = rf_age_category.score(x_test_tfidf, y_test['age_category'])
print(f'Random Forest age category prediction accuracy: {age_category_score}')
```

Calculation of metrics for the age model and feature selection:

```

age_category_predictions = rf_age_category.predict(x_test_tfidf)
print(classification_report(y_test['age_category'], age_category_predictions))
importances_age_category = rf_age_category.feature_importances_
indices_age_category = np.argsort(importances_age_category)[::-1]
top_n = 15
top_features_age_category = [vectorizer.get_feature_names_out()[i] for i in
indices_age_category[:top_n]]
print(f'Top {top_n} features for age category prediction: {top_features_age_category}')

```

Model for gender determination:

```

rf_gender = RandomForestClassifier(n_estimators=100, random_state=42)
rf_gender.fit(x_train_tfidf, y_train['gender_bi'])
rf_gender.score(x_test_tfidf, y_test['gender_bi'])
gender_predictions = rf_gender.predict(x_test_tfidf)
print(classification_report(y_test['gender_bi'], gender_predictions))

```

Selection of features for the gender model:

```

importances_gender = rf_gender.feature_importances_
indices_gender = np.argsort(importances_gender)[::-1]
top_features_gender = [vectorizer.get_feature_names_out()[i] for i in indices_gender[:top_n]]
print(f'Top {top_n} features for gender prediction: {top_features_gender}')

```

Visualization of graphs of the importance of traits for age:

```

top_importances_age_category = importances_age_category[indices_age_category[:top_n]]
plt.figure(figsize=(10, 6))
plt.barh(range(top_n), top_importances_age_category, align='center', color='salmon')
plt.yticks(range(top_n), top_features_age_category)
plt.gca().invert_yaxis()
plt.xlabel('Feature Importance')
plt.title('Top 15 Features for Age Category Prediction')
plt.show()

```

Visualization of graphs of the importance of traits for gender:

```

top_importances_gender = importances_gender[indices_gender[:top_n]]
plt.figure(figsize=(10, 6))
plt.barh(range(top_n), top_importances_gender, align='center', color='purple')
plt.yticks(range(top_n), top_features_gender)
plt.gca().invert_yaxis()
plt.xlabel('Feature Importance')
plt.title('Top 15 Features for Gender Prediction')
plt.show()

```

Correlation matrix for gender predictions:

```

from sklearn.metrics import confusion_matrix
import seaborn as sns
predicted_genders = rf_gender.predict(x_test_tfidf)
cm = confusion_matrix(y_test['gender_bi'], predicted_genders)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='RdPu', xtickLabels=['Female', 'Male'],
ytickLabels=['Female', 'Male'])
plt.title('Confusion Matrix for Gender Prediction')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()

```

Correlation matrix for age predictions:

```

cm_age_category = confusion_matrix(y_test['age_category'], age_category_predictions)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_age_category, annot=True, fmt='d', cmap='RdPu', xtickLabels=['<20', '20-30',
'>30'], ytickLabels=['<20', '20-30', '>30'])
plt.title('Confusion Matrix for Age Category Prediction')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()

```

## Derivation of ROC curves for gender and age:

```
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import label_binarize
from itertools import cycle
y_test_binarized = label_binarize(y_test['age_category'], classes=[0, 1, 2])
n_classes = y_test_binarized.shape[1]
fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test_binarized[:, i],
rf_age_category.predict_proba(x_test_tfidf)[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
    fpr_gender, tpr_gender, _ = roc_curve(y_test['gender_bi'],
rf_gender.predict_proba(x_test_tfidf)[:, 1])
    roc_auc_gender = auc(fpr_gender, tpr_gender)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))
colors = cycle(['darkturquoise', 'darkmagenta', 'lightcoral'])
for i, color in zip(range(n_classes), colors):
    ax1.plot(fpr[i], tpr[i], color=color, lw=2, label='ROC curve of class {0} (area =
{1:0.2f})'.format(i, roc_auc[i]))
    ax1.plot([0, 1], [0, 1], 'k--', lw=2)
    ax1.set_xlim([0.0, 1.0])
    ax1.set_ylim([0.0, 1.05])
    ax1.set_xlabel('False Positive Rate')
    ax1.set_ylabel('True Positive Rate')
    ax1.set_title('ROC Curve for Age Category Prediction')
    ax1.legend(loc="lower right")
    ax2.plot(fpr_gender, tpr_gender, color='purple', lw=2, label='ROC curve (area =
{0:0.2f})'.format(roc_auc_gender))
    ax2.plot([0, 1], [0, 1], 'k--', lw=2)
    ax2.set_xlim([0.0, 1.0])
    ax2.set_ylim([0.0, 1.05])
    ax2.set_xlabel('False Positive Rate')
    ax2.set_ylabel('True Positive Rate')
    ax2.set_title('ROC Curve for Gender Prediction')
    ax2.legend(loc="lower right")
plt.tight_layout()
plt.show()
```

## Derivation of the precision-recall curve:

```
from sklearn.metrics import precision_recall_curve
precision = dict()
recall = dict()
for i in range(n_classes):
    precision[i], recall[i], _ = precision_recall_curve(y_test_binarized[:, i],
rf_age_category.predict_proba(x_test_tfidf)[:, i])
plt.figure(figsize=(8, 6))
for i, color in zip(range(n_classes), colors):
    plt.plot(recall[i], precision[i], color=color, lw=2, label='PR curve of class
{0}'.format(i))
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve for Age Category Prediction')
plt.legend(loc="lower left")
plt.show()
```

## Graph comparing actual and predicted age:

```
plt.figure(figsize=(12, 6))
sns.countplot(x=y_test['age_category'], palette='Blues', alpha=0.5, label='True Age
Categories')
```

```

sns.countplot(x=age_category_predictions, palette='Reds', alpha=0.5, Label='Predicted Age
Categories')
plt.title('Distribution of True vs. Predicted Age Categories')
plt.xlabel('Age Category')
plt.ylabel('Count')
plt.legend()
plt.show()

```

A function for converting predicted data into a human-readable format:

```

def decategorize(age:list, gender:list):
    res = []
    if age == 0:
        res.append(['less than 20'])
    elif age == 1:
        res.append(['20-30'])
    else:
        res.append(['more than 30'])
    if gender == 0:
        res.append(['female'])
    else:
        res.append(['male'])
    return res

```

Downloading the second dataset (with book authors):

```
book_train = pd.read_csv('train_book.csv')
```

Conversion of author names to gender and pre-processing of text data using previously described functions:

```

book_train.drop(columns=['id'])
def categorize_gender(gender): # defining authors gender
    if gender == 'EAP' or gender == 'HPL':
        return 1 # male
    else:
        return 0 # female
# Apply age categorization
book_train['gender'] = book_train['author'].apply(categorize_gender)
book_train['text'].apply(text_prerocess)
book_train['text'].apply(stopwords_removal)
x_train_book, x_test_book, y_train_book, y_test_book = train_test_split(book_train['text'],
book_train[['gender']], test_size=0.2, random_state=42)

```

Data vectorization:

```

x_train_b = vectorizer.fit_transform(x_train_book)
x_test_b = vectorizer.transform(x_test_book)

```

Training the additional model purely on new data:

```

rf_author = RandomForestClassifier(n_estimators=100, random_state=42)
rf_author.fit(x_train_b, y_train_book)
rf_author.score(x_test_b, y_test_book)

```

Comparison of the results of the old and new gender models on the old and new data:

```

print('Old model on old data: ', rf_gender.score(x_test_tfidf, y_test['gender_bi']))
print('Old model on new data: ', rf_gender.score(x_test_b, y_test_book))
print('New model on new data: ', rf_author.score(x_test_b, y_test_book))
print('New model on old data: ', rf_author.score(x_test_tfidf, y_test['gender_bi']))

```

String to enter sample text from the user:

```
your_text = 'I would love to go to the gallery with you after university! I heard they are
exhibiting the most famous paintings of Monet, he is my favourite painter, i am so excited!'
```

Functions for converting text, making predictions, converting predictions back to text:

```

text = text_prerocess(your_text) #Now run this cell and it will give you result
text = stopwords_removal(text)
text = pd.Series(your_text)
text = vectorizer.transform(text)
pred_age = rf_age_category.predict(text)

```

```

pred_gen = rf_gender.predict(text)
result = decategorize(pred_age, pred_gen)
print(f'The author\'s age is in this range: {result[0]}\nAuthor\'s gender: {result[1]}')

```

## 7. Results

The program is divided into several sections according to the type of tasks performed. When starting the program, you first need to import the libraries, immediately after that the data analysis section begins.

```

import numpy as np
import pandas as pd
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer

Data analysis

df = pd.read_csv('./blogtext.csv')
df.head()

   id  gender  age  topic  sign  date  text
0  2059027  male   15  Student  Leo  14,May,2004  Info has been found (+/- 100 pages,...
1  2059027  male   15  Student  Leo  13,May,2004  These are the team members: Drewe...
2  2059027  male   15  Student  Leo  12,May,2004  In het kader van kernfusie op aarde...
3  2059027  male   15  Student  Leo  12,May,2004  testing!!! testing!!!
4  3581210  male   33  InvestmentBanking  Aquarius  11,June,2004  Thanks to Yahoo!'s Toolbar I can ...

df = df.drop(columns=['id', 'date', 'sign']) #deleting unnessesary columns

```

Figure 13: Start the program

In the Data Analysis section, the dataset has been imported (if necessary, the reference to the dataset in the file system must be changed) and unnecessary data columns (id, sign, date) have been removed. After that, the data balance in the dataset was checked using data visualization. A quantity graph was used to compare gender, and a histogram was used to compare age.

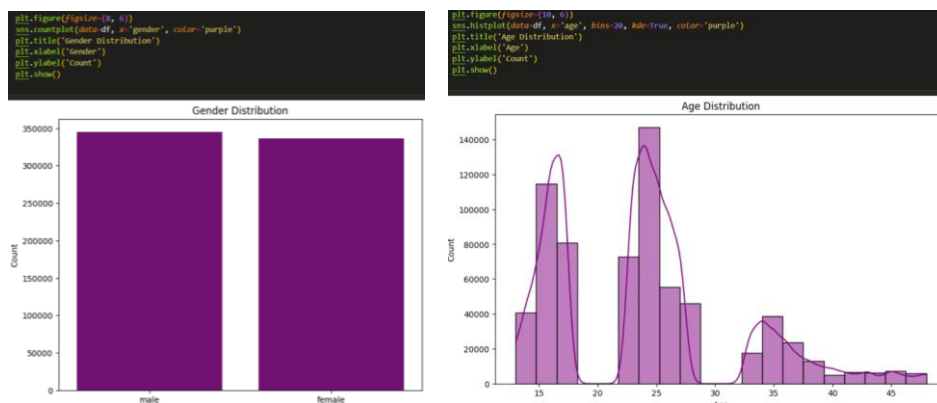


Figure 14: Data balance graphs

Next, data pre-processing takes place. This section creates two important functions that are responsible for cleaning data, as well as one optional function:

- `def stopwords_removal(text)`: the function removes stop words, removes capital letters, tokenizes sentences and provides lemmatized text. As input, it accepts the text to be cleared, and returns the cleared text.

```
lemmatizer = WordNetLemmatizer()

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')

def stopwords_removal(text):
    new_words = word_tokenize(text)
    new_filtered_words = [
        #lemmatizer.lemmatize(word.lower()) for word in new_words if word.lower() not in stopwords.words('english')
        word for word in new_words if word.lower() not in stopwords.words('english')
    ]
    return ' '.join(new_filtered_words)
```

**Figure 15:** Stopwords\_removal function

- `def text_prerocess(text)`: function for text pre-processing, it removes characters and unwanted combinations that can cause noise. As input, it accepts the text to be cleared, returns the cleared text.

```
def text_prerocess(text): #10 minutes
    text = re.sub(r'<.*>', '', text)
    text = re.sub(r'\W+', ' ', text)
    text = text.lower()
    text = re.sub(r'\n', ' ', text)
    text = re.sub(r'\urlink', '', text)
    text = re.sub(r'im', 'i am', text)

    return text
```

```
def categorize_age(age):
    if age < 20:
        return 0 # less than 20
    elif 20 <= age <= 30:
        return 1 # 20-30
    else:
        return 2 # more than 30

df_short['age_category'] = df_short['age'].apply(categorize_age)
```

**Figure 16:** The text\_prerocess function      **Figure 17:** Age categorization function

- `def categorize_age(age)`: an optional function, used only when age is defined as a classification by age group rather than assumed as a number. Divide authors into three groups by age. At the entrance, it takes the age, at the exit it gives the number of the group to which the author belongs.

In general, in this section, the text data is passed through these functions in turn to obtain tokenized and cleaned text, which can then be immediately vectorized and sent to the model.

```
df_short['clean_text'] = df_short['text'].apply(text_prerocess)
df_short['clean_topic'] = df_short['topic'].apply(text_prerocess)

df_short['clean_topic'] = df_short['clean_topic'].apply(stopwords_removal)
df_short['clean_text'] = df_short['clean_text'].apply(stopwords_removal)

df_short['combined_text'] = df_short['clean_text'] + ' ' + df_short['clean_topic']

df_short['gender_bi'] = df_short['gender'].map({'male': 1, 'female': 0})

x_train, x_test, y_train, y_test = train_test_split(df_short['combined_text'], df_short[['age_category', 'gender_bi']], test_size=0.2, random_state=42)
```

**Figure 18:** Applying functions to data



In the next section, the model itself, or rather two models, is trained to predict age and gender. However, the data from the previous section is still in text form, which is incomprehensible to the model, so the text is vectorized before that. It converts the text into a set of embeddings, in the form of TF-IDF, a technology that allows you to sort words by their importance, which is very important for my research.

```
vectorizer = TfidfVectorizer(max_features=10000, ngram_range=(1, 2))
x_train_tfidf = vectorizer.fit_transform(x_train)
x_test_tfidf = vectorizer.transform(x_test)
```

**Figure 19:** Text vectorization for model training

The vectorized ones are passed to the RandomForestClassifier model to determine gender or age in a categorical form, if the age needs to be predicted in a numerical form, the RandomForestRegressor model is used. These models were chosen because they allow us to highlight features for further analysis.

```
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.linear_model import LinearRegression, Ridge, Lasso

# Train a Random Forest classifier for age prediction
rf_age = RandomForestRegressor(max_depth=16, max_leaf_nodes=32)
rf_age.fit(x_train_tfidf, y_train['age'])

rf_age.score(x_test_tfidf, y_test['age'])
✓ 1m 33.1s
0.24613097157880148

from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier, RandomForestRegressor
from sklearn.linear_model import LinearRegression, Ridge, Lasso

rf_age_category = RandomForestClassifier(n_estimators=100, random_state=42)
rf_age_category.fit(x_train_tfidf, y_train['age_category'])

age_category_score = rf_age_category.score(x_test_tfidf, y_test['age_category'])
print(f'Random Forest age category prediction accuracy: {age_category_score}')

Random Forest age category prediction accuracy: 0.64325

rf_gender = RandomForestClassifier(n_estimators=100, random_state=42)
rf_gender.fit(x_train_tfidf, y_train['gender_bi'])
💡
rf_gender.score(x_test_tfidf, y_test['gender_bi'])

gender_predictions = rf_gender.predict(x_test_tfidf)
print(classification_report(y_test['gender_bi'], gender_predictions))
```

	precision	recall	f1-score	support
0	0.63	0.73	0.68	2019
1	0.67	0.56	0.61	1981

**Figure 20:** The models used in the work

Also, in this section, the characteristics that have the greatest influence on the classification result are highlighted.

```

age_category_predictions = rf_age_category.predict(x_test_tfidf)
print(classification_report(y_test['age_category'], age_category_predictions))

importances_age_category = rf_age_category.feature_importances_
indices_age_category = np.argsort(importances_age_category)[::-1]

top_n = 15
top_features_age_category = [vectorizer.get_feature_names_out()[i] for i in indices_age_category[:top_n]]
print(f'Top {top_n} features for age category prediction: {top_features_age_category}')

importances_gender = rf_gender.feature_importances_
indices_gender = np.argsort(importances_gender)[::-1]

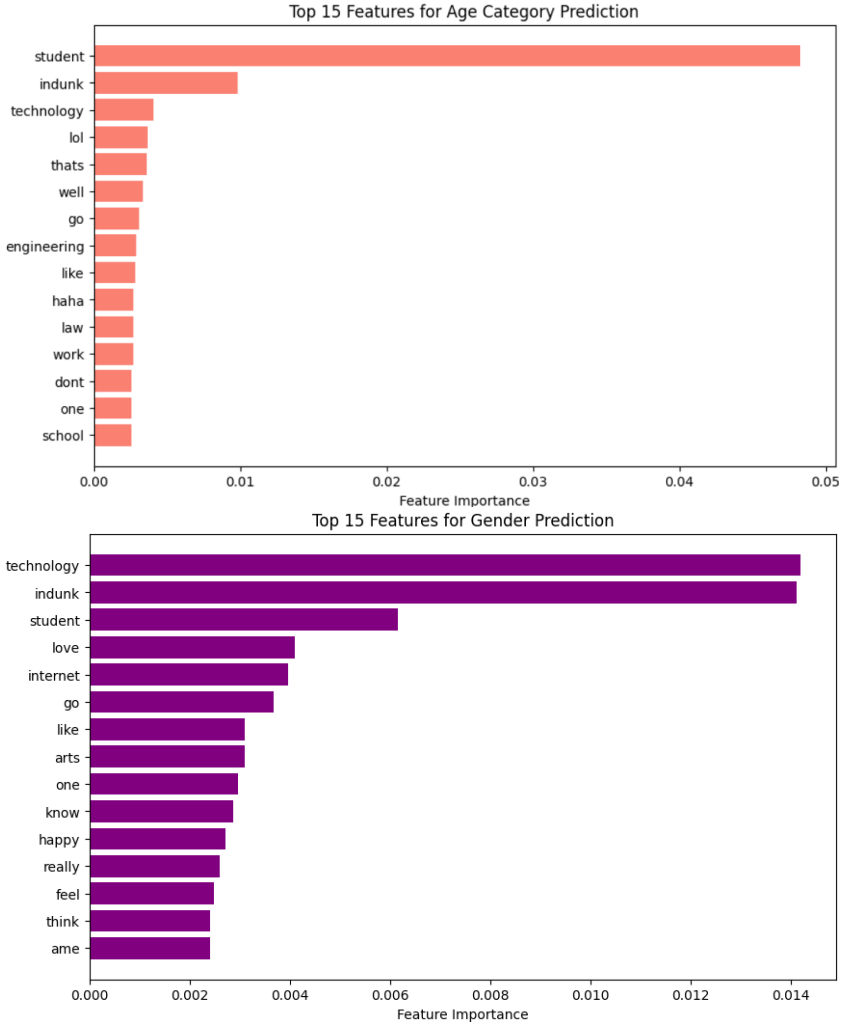
top_features_gender = [vectorizer.get_feature_names_out()[i] for i in indices_gender[:top_n]]
print(f'Top {top_n} features for gender prediction: {top_features_gender}')

Top 15 features for gender prediction: ['technology', 'indunk', 'student', 'love', 'internet', 'know', 'go', 'like'

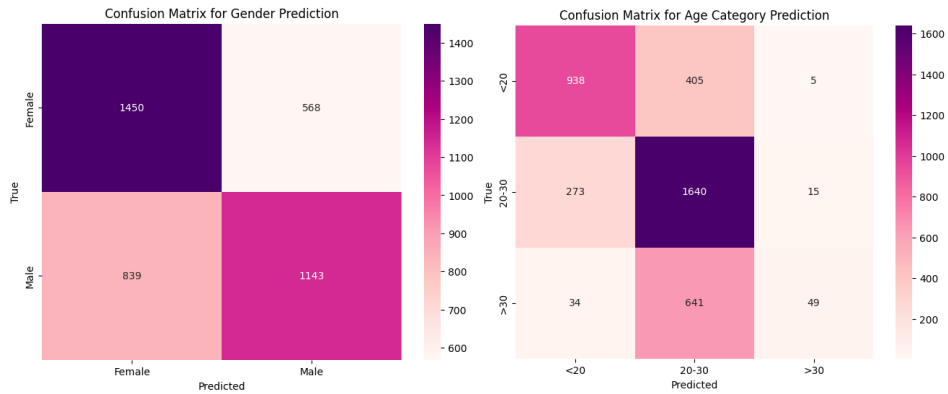
```

**Figure 21:** Extracting the main features

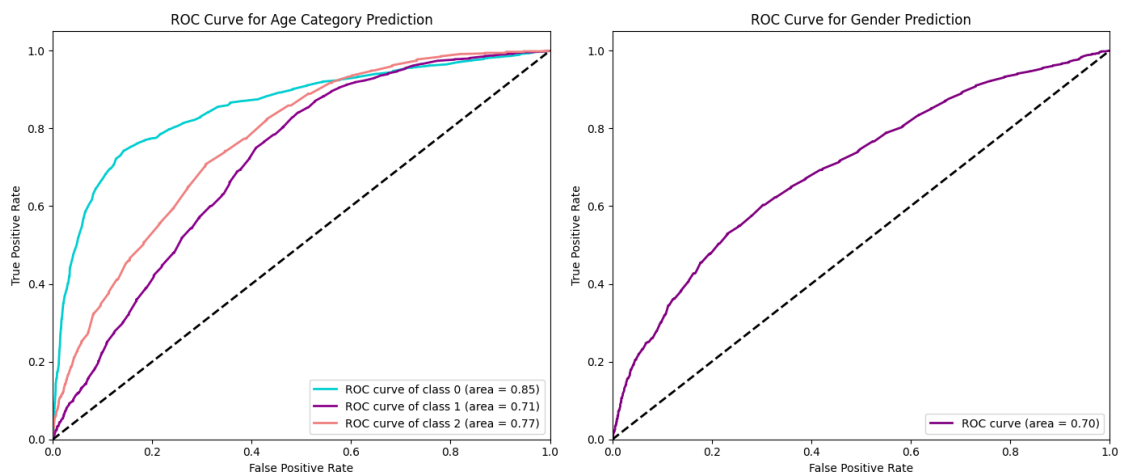
In the next section, the data obtained during the research is analysed. The section consists entirely of graphs of various types and for various purposes.



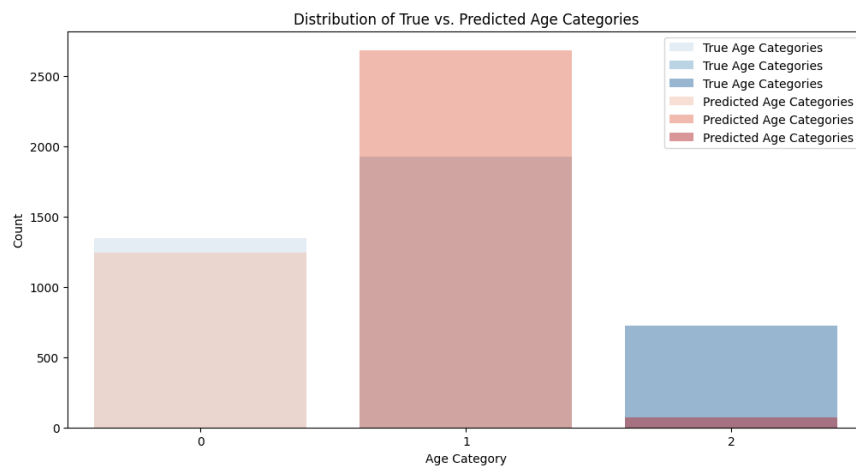
**Figure 22:** Ranking of signs that affect the determination of gender and age



**Figure 23:** Correlation matrices of age (categories) and gender



**Figure 24:** ROC curves



**Figure 25:** Comparison of predicted categories with actual ones

A separate section has also been developed for this project so that users have the opportunity to more conveniently interact with the model and enter their sentences for

age and gender verification. For the sake of the experiment, 5 different sentences were given: 3 from women and 2 from men. The following results were obtained:

```

#please enter your text below:
your_text = 'I would love to go to the gallery with you after university! I heard they are exhibiting the most famous paintings of Monet, he is my favourite painter, i am so excited
#your_text = 'But actually I can just dye my hair, wear my tights and clothes with crazy colors, and bam, ready' #me
#your_text = 'I'm still a bit mad at myself for not fixing my bike correctly' #male 22
#your_text = 'During the last week, I was busy with university things, volunteering project tasks and job I am doing for living. I am overwhelmed and exhausted, I would like to get
#your_text = 'I finnaly found satated CAP for GPT-4o. Interesting that new technology has bigger cap than old one, but new one is also better.' #male 21
0.0s Python

#Now run this cell and it will give you result

text = text_prerocess(your_text)
text = stopwords_removal(text)

text = pd.Series(your_text)

text = vectorizer.transform(text)

pred_age = rf_age_category.predict(text)
pred_gen = rf_gender.predict(text)

result = decategorize(pred_age, pred_gen)
print(f'The author\'s age is in this range: {result[0]}\nAuthor\'s gender: {result[1]}')
✓ 0.0s
The author's age is in this range: ['20-30']
Author's gender: ['female']

```

Figure 26: Results of the second case

```

#please enter your text below:
#your_text = 'I would love to go to the gallery with you after university! I heard they are exhibiting the most famous paintings of Monet, he is my favourite painter, i am so excit
your_text = 'But actually I can just dye my hair, wear my tights and clothes with crazy colors, and bam, ready' #me
#your_text = 'I'm still a bit mad at myself for not fixing my bike correctly' #male 22
#your_text = 'During the last week, I was busy with university things, volunteering project tasks and job I am doing for living. I am overwhelmed and exhausted, I would like to get
#your_text = 'I finnaly found satated CAP for GPT-4o. Interesting that new technology has bigger cap than old one, but new one is also better.' #male 21
✓ 0.0s Python

#Now run this cell and it will give you result

text = text_prerocess(your_text)
text = stopwords_removal(text)

text = pd.Series(your_text)

text = vectorizer.transform(text)

pred_age = rf_age_category.predict(text)
pred_gen = rf_gender.predict(text)

result = decategorize(pred_age, pred_gen)
print(f'The author\'s age is in this range: {result[0]}\nAuthor\'s gender: {result[1]}')
✓ 0.0s
The author's age is in this range: ['20-30']
Author's gender: ['female']

```

Figure 27: Results of the second case

```

#please enter your text below:
#your_text = 'I would love to go to the gallery with you after university! I heard they are e
#your_text = 'But actually I can just dye my hair, wear my tights and clothes with crazy colo
your_text = 'I\'m still a bit mad at myself for not fixing my bike correctly' #male 22
#your_text = 'During the last week, I was busy with university things, volunteering project t
#your_text = 'I finnaly found satated CAP for GPT-4o. Interesting that new technology has big
✓ 0.0s

#Now run this cell and it will give you result

text = text_prerocess(your_text)
text = stopwords_removal(text)

text = pd.Series(your_text)

text = vectorizer.transform(text)

pred_age = rf_age_category.predict(text)
pred_gen = rf_gender.predict(text)

result = decategorize(pred_age, pred_gen)
print(f'The author\'s age is in this range: {result[0]}\nAuthor\'s gender: {result[1]}')
✓ 0.0s
The author's age is in this range: ['20-30']
Author's gender: ['male']

```

Figure 28: Results of the third case

```

e to go to the gallery with you after university! I heard they are exhibiting the most famous paintings of Monet, he is my favourite painter, I am so excited! #me
y I can just dye my hair, wear my tights and clothes with crazy colors, and bam, ready' #me
a bit mad at myself for not fixing my bike correctly' #male 22
ast week, I was busy with university things, volunteering project tasks and job I am doing for living. I am overwhelmed and exhausted, I would like to get enough sleep' #male 21
ound satated CAP for GPT-4o. Interesting that new technology has bigger cap than old one, but new one is also better.' #male 21

```

✓ 0.0s

```

#Now run this cell and it will give you result

text = text_preprocess(your_text)
text = stopwords_removal(text)

text = pd.Series(your_text)

text = vectorizer.transform(text)

pred_age = rf_age_category.predict(text)
pred_gen = rf_gender.predict(text)

result = decategorize(pred_age, pred_gen)
print(f'The author\'s age is in this range: {result[0]}\nAuthor\'s gender: {result[1]}')

```

✓ 0.0s

```

The author's age is in this range: ['20-30']
Author's gender: ['male']

```

**Figure 29:** Results of the fourth case

```

#please enter your text below:
#your_text = 'I would love to go to the gallery with you after university! I heard they are exhibiting the most famous paintings of Monet, he is my favou
#your_text = 'But actually I can just dye my hair, wear my tights and clothes with crazy colors, and bam, ready' #me
#your_text = 'I\'m still a bit mad at myself for not fixing my bike correctly' #male 22
#your_text = 'During the last week, I was busy with university things, volunteering project tasks and job I am doing for living. I am overwhelmed and exh
your_text = 'I finnaly found satated CAP for GPT-4o. Interesting that new technology has bigger cap than old one, but new one is also better.' #male 21

```

✓ 0.0s

```

#Now run this cell and it will give you result

text = text_preprocess(your_text)
text = stopwords_removal(text)

text = pd.Series(your_text)

text = vectorizer.transform(text)

pred_age = rf_age_category.predict(text)
pred_gen = rf_gender.predict(text)

result = decategorize(pred_age, pred_gen)
print(f'The author\'s age is in this range: {result[0]}\nAuthor\'s gender: {result[1]}')

```

✓ 0.0s

```

The author's age is in this range: ['20-30']
Author's gender: ['male']

```

**Figure 30:** Results of the fifth case

Four of the five proposed cases are correctly identified. One of the woman's messages is identified as a message from a man.

## 8. Discussion

So, the purpose of the research was to create a model for analysing the author's age and gender based on his texts. In this work, the model is built and trained on a part of the dataset with blogs. The following program execution results were achieved:

- The age determination model demonstrates an accuracy of 64.3% if age is specified categorically, and 25% if age is assumed in numerical format. The sex determination model has an accuracy of 64%. These results are not as high as

we would like, but they allow us to extract certain features from the text that allow us to determine the biological data of the author. The data could probably be improved by taking a larger dataset and longer messages.

- Features from the model were selected and sorted, and a rating of the most influential features for the classification of biological data was obtained. Adding the subject of writing to the overall text of the message greatly helps the classification, making it easier for models to predict both age and gender. For age classification, topics about the place of study or work, and social status (Student, industry, and others) are the most helpful, for gender it is the field of interests (technology, finance).
- From the correlation matrix, it can be noted that men's messages are perceived as women's more often than vice versa. This may indicate that men are more likely to use a feminine way of communicating, or that women are more likely to indicate their gender as the opposite for various reasons.
- When analysing a dataset from books, where it was necessary to determine the gender of the author, an additional model was created, which was trained only on this dataset, and also passed this dataset through the previous model and vice versa. The results of operations were compared. It turns out that a model trained on blog data performs better on unfamiliar data than a model trained on new data.

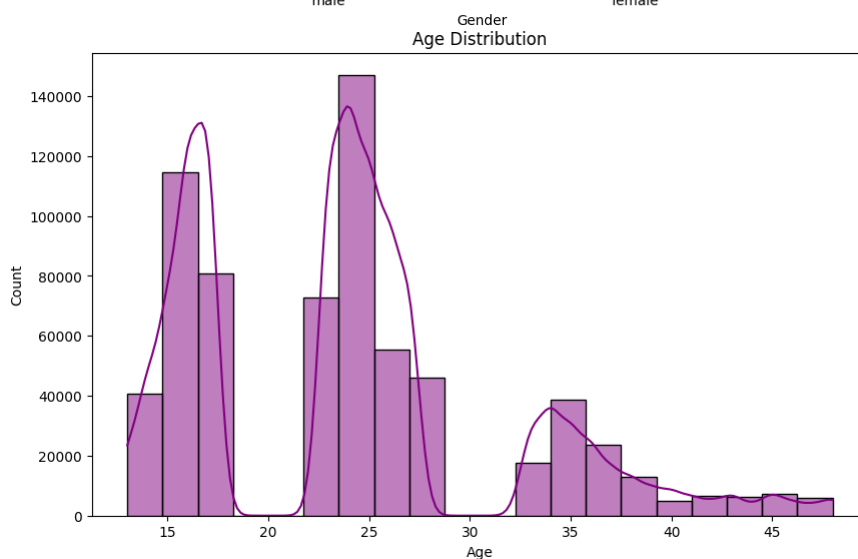
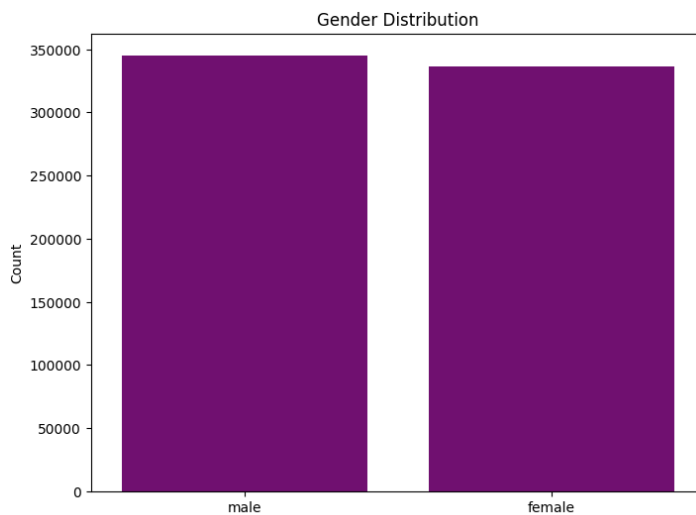
```
print('Old model on old data: ', rf_gender.score(x_test_tfidf, y_test['gender_bi']))
print('Old model on new data: ', rf_gender.score(x_test_b, y_test_book))
print('New model on new data: ', rf_author.score(x_test_b, y_test_book))
print('New model on old data: ', rf_author.score(x_test_tfidf, y_test['gender_bi']))
✓ 0.8s
Old model on old data: 0.643
Old model on new data: 0.6223186925434117
New model on new data: 0.8951583248212462
New model on old data: 0.50775
```

**Figure 31:** Comparison of two models

- One of the authors wrote a message from himself, as an example, which is correctly classified as female, age 20 to 30, which it is. The numerical value of the age slightly exaggerates the real one, the model believes that the author is 27. It can be said that the model is much more likely to determine the psychological age of a person than the biological one (the author is a little over 20 years old).

Statistics of the model before training:

1. Statistical analysis of data begins even before sending the dataset to the model for making predictions. When the dataset is loaded, two graphical displays of the data distribution in this dataset are constructed. The amount of data by gender is almost the same, but the ages 13-17 and 23-29 years old are significantly more prevalent, and there are no users in their 20s and 30s at all. This can have a rather negative effect on the accuracy of the model. Blog Authorship Corpus dataset (kaggle.com) [73].



**Figure 32:** Graphs of age and gender distribution

- During the training of the model, to determine its accuracy, a classification report is built, which includes such metrics as accuracy, f1-score, macro avg, precision, recall.

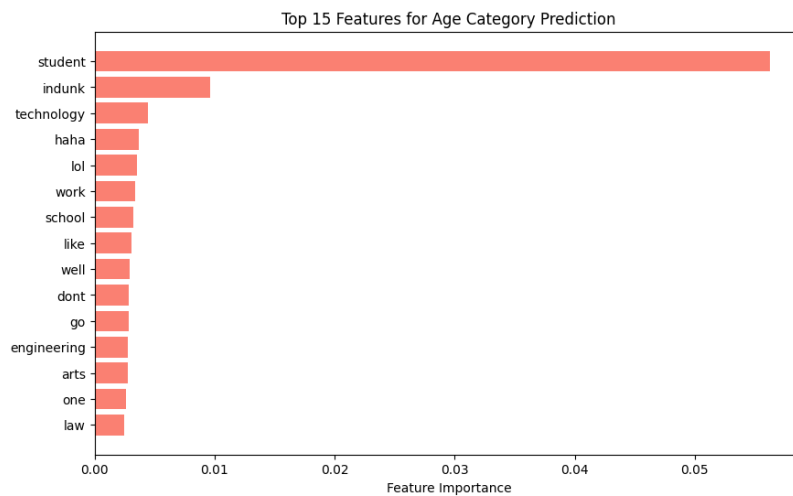
	precision	recall	f1-score	support
0	0.74	0.68	0.71	2097
1	0.59	0.83	0.69	2830
2	0.67	0.07	0.13	1073
accuracy			0.64	6000
macro avg	0.67	0.53	0.51	6000
weighted avg	0.66	0.64	0.60	6000

	precision	recall	f1-score	support
0	0.63	0.72	0.67	2967
1	0.68	0.58	0.62	3033
accuracy			0.65	6000
macro avg	0.65	0.65	0.65	6000
weighted avg	0.65	0.65	0.65	6000

**Figure 33:** Classification Report for Age(1) and Gender(2)

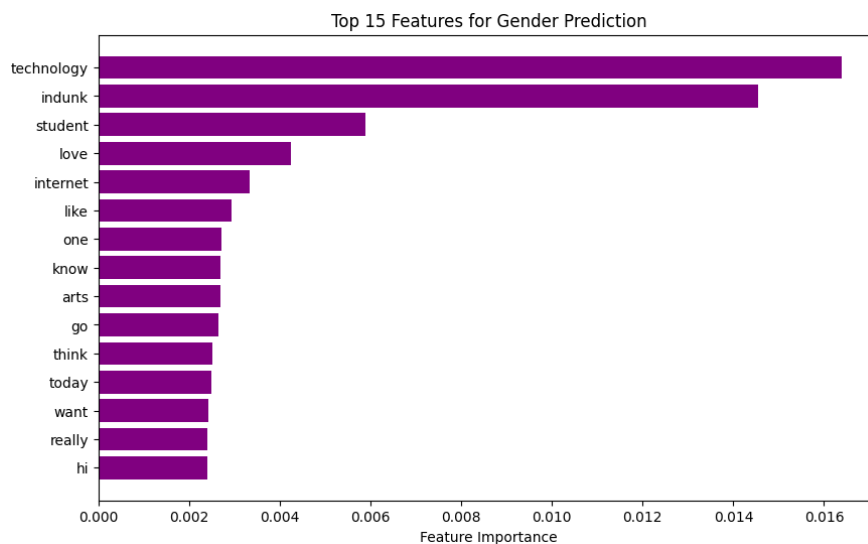
Gender determination has more stable metrics, it can be assumed that models are easier to cope with gender determination than age.

- Analysis of signs of predicting gender or age is perhaps the most important step in this research. The results were generated and displayed in the form of a ranked graph of the 15 most important features for classification.



**Figure 34:** Signs for predicting age

The first three places belong to the topic names that users specify when writing a blog. This is quite expected because the subject of the text can tell about the user no more than the text itself. The following positions belong to slang forms of the text, which, most likely, are more often used by young people. Next, you can see that places of work/study and some other words appear, the meaning of which can be analysed by analogy.

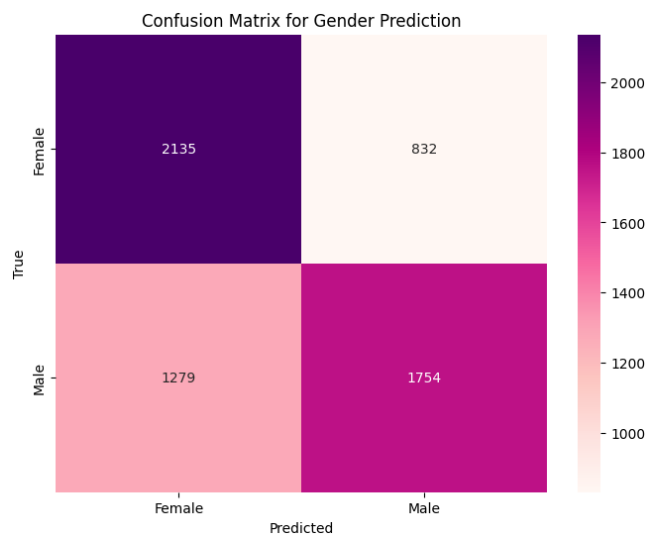


**Figure 35:** Signs for predicting gender



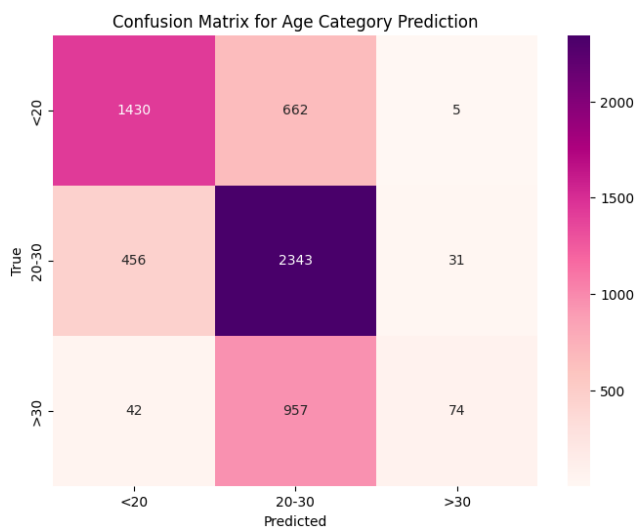
Again, the first to appear are the topics of messages that indicate the areas of interest of individuals. More technical interests are responsible for the male gender, while more creative interests are for the female gender. You can also notice that there are quite a lot of feelings, the expression of which is more characteristic of women.

4. Correlation matrices can also be a great source of information about a program and its results.



**Figure 36:** Correlation matrix for gender

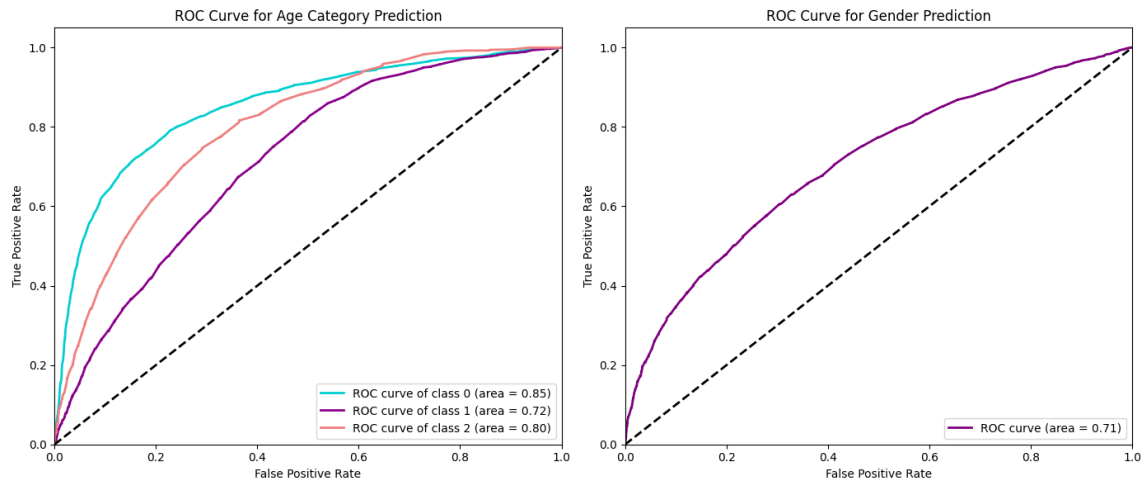
In many previous runs of the program, it can be observed that the number of men classified as women is relatively higher than the other way around. From this, we can draw an interesting conclusion that men more often adopt a female manner of communication than women - a male one.



**Figure 37:** Correlation matrix for age (classification)

It can be seen from the matrix that the central group quite strongly dominates the others, both in the classification of the model and in the dataset. This problem can be solved by age-balancing the dataset, which may greatly distort the model results.

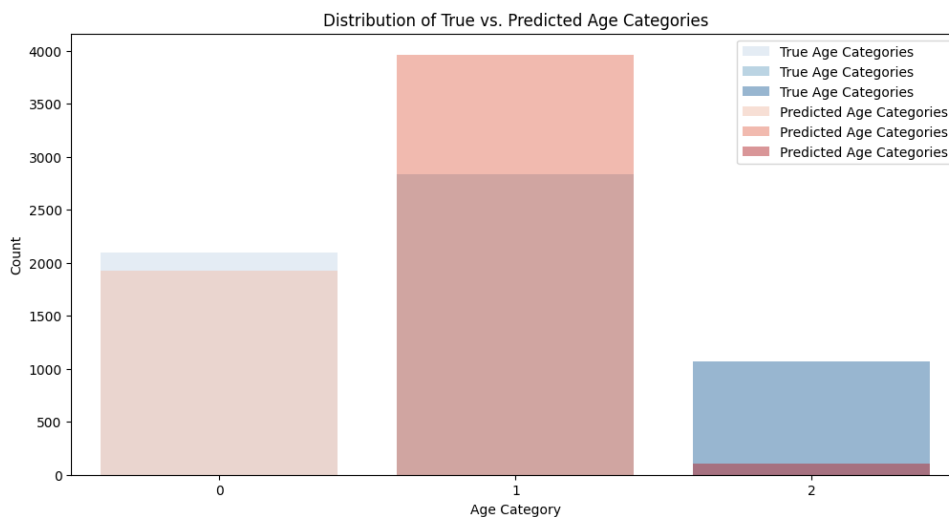
- The ROC curve shows the ratio of True Positive to True Negative and the accuracy with which the model works.



**Figure 38:** Curves for age and gender

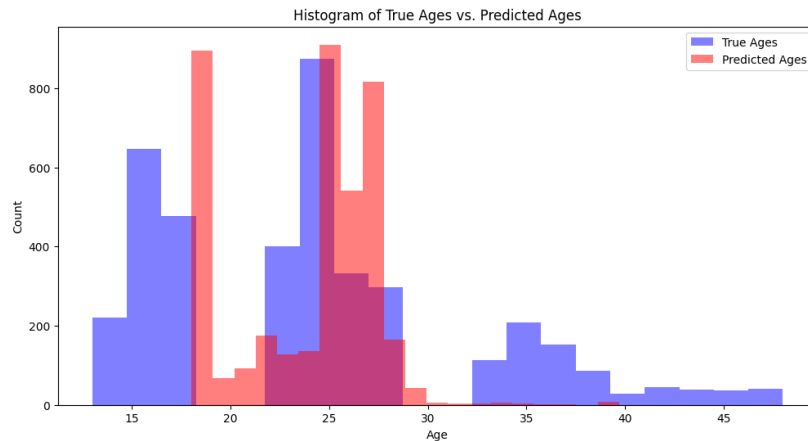
It can be seen that the model predicts the age group <20 best, although it is not the most common. Most likely, many signs help to quickly identify this age, such as "school" or "student".

- Comparing predicted and actual ages is one of the best ways to visually evaluate model predictions.



**Figure 39:** Comparison of predicted and actual categories (age)

As you can see, the model has difficulties with the definition of the third category (>30). Most likely, such trouble is because the number of this category, compared to others, is very small, so the model has not learned how to predict it properly. This can fix the correction of the dataset. You can still look at a similar graph but for a temporary prediction of age.



**Figure 40:** Comparison of predictions and true values (age)

Interestingly, the model does not want to predict ages below 17, even though there are many younger users in the dataset. It predicts a large part of the data as 17-23, although it is generally an empty area in the dataset. Also, the model, as in the categories, practically avoids age more than 30. It seems that more weighted data will be able to solve some of these problems.

7. A separate gender classification model has been developed for the dataset of book authors, which perfectly classifies the gender of authors from this dataset. Dataset with book authors: Spooky Author Identification [74].

**0.8051583248212462**

**Figure 41:** Prediction of the "book" model

However, when this same data was run through a previously trained model on the old data, it performed better than this new model on the old data. So, the old model, although it has worse accuracy, perceives the new data much better than the new model.

```

print('Old model on old data: ', rf_gender.sc
print('Old model on new data: ', rf_gender.sc
print('New model on new data: ', rf_author.sc
print('New model on old data: ', rf_author.sc
✓ 0.8s
Old model on old data: 0.643
Old model on new data: 0.6223186925434117
New model on new data: 0.8051583248212462
New model on old data: 0.50775

```

**Figure 42:** Comparison of models

- As a final test, as well as an opportunity for users to more conveniently interact with the program, the ability to determine the gender and age of the text specified by the user has been developed. A sentence was written by one of the authors of this work, and a fairly accurate classification was obtained.

```
✓ 0.0s
The author's age is in this range: ['20-30']
Author's gender: ['female']
```

**Figure 43:** Predictions of the model

Several experiments were also conducted to verify the model. Three friends of one of the authors of the study took part in the experiment. They wrote common conversational sentences, and this author also wrote a few. These sentences were placed in the following order: author (female, 21), author (female, 21), male 22, female 20, male 21. The obtained results are arranged in the same order.

```
The author's age is in this range: ['20-30']
Author's gender: ['female']
The author's age is in this range: ['20-30']
Author's gender: ['female']
The author's age is in this range: ['20-30']
Author's gender: ['male']
The author's age is in this range: ['20-30']
Author's gender: ['male']
The author's age is in this range: ['20-30']
Author's gender: ['male']
```

**Figure 44:** Results of predictions for entered sentences

Four out of five sentences are classified correctly, and all of them are correctly determined by age, according to article 4/5. One woman was classified as a man. This is a pretty good result for a text-based age and gender prediction model.

A few more experiments were conducted with sentence formulation using the features provided by the model. Thus, adding the words "school" or "student" reduces the predicted age of the author, adding words related to technology changes the gender of the author to male. This means that it is important to submit a sentence to the model that is not written to deceive the model, it should be sincere and casual.

```
#your_text = 'I finally found satated CAP for GPT-4o. Interesting that new technology has bigger cap than old one, but new one
your_text = 'studies But actually I can just school dye my hair, wear my tights and clothes with crazy colors, and bam, ready'
✓ 0.0s
The author's age is in this range: ['less than 20']
Author's gender: ['female']
```

**Figure 45:** Artificial reduction of predicted age

## 9. Conclusions

So, in the process of implementing this project, namely the project on determining the author's age and gender based on his text, a model was developed that determines these biological data of the author based on his text. Before starting work, similar studies on a similar topic are reviewed to find out what has already been researched and tested, and what is still worth investigating. Also, from these studies, it was possible to find many hints about which implementation methods and tools are better to choose, and which work better for this task.

The work on the project is carefully planned using process diagrams and data flows. The best methods and tools for the implementation of this project were studied, and simple classification and regression models of Random Forest became such tools. Such models were chosen, because they cope with the task quite well, and are much less resource-intensive than the same large language models, in addition, they are very easy to use and configure.

Two datasets were selected, a dataset with blogs and a dataset with books. The dataset with blogs was used the most because it contains both the age and gender of the blog author.

Before use, the data was analysed and cleaned, later transformed into embeddings and sent for model training. The results of the model are studied and analysed in detail. Many useful features are extracted that are responsible for classifying the age or gender of the author in the texts. In addition, many interesting regularities were observed in the process of analysing the results. Additionally, a test case is implemented that allows the user to easily interact with my model.

Such research is very useful in many areas of life, but also for the development of science. Such studies can help capture the relationship between seemingly unrelated features, such as the reflection of an author's gender and age in his texts. We believe that it is possible to try to repeat or edit our experiment on a computer with higher capacities to be able to analyse much larger volumes of data, which could significantly improve the results of the model. Although it is worth noting, as stated in another study, such predictions work more for the psychological age of a person than for his biological age, because the manner of speaking reflects the psychological age. Also, you can try using other datasets in the future, if available, or rebalance the current dataset and try again. Such research can bring many benefits to society if it is used properly.

## References

- [1] O. Tverdokhlib, V. Vysotska, P. Pukach, M. Vovk, Information technology for identifying hate speech in online communication based on machine learning, *Lecture Notes on Data Engineering and Communications Technologies* 195 (2024) 339–369.
- [2] N. Borysova, K. Melnyk, N. Babkova, Z. Kochuieva, V. Melnyk, Gender Classification of Surnames: Ukrainian aspect, *CEUR Workshop Proceedings* 3171 (2022) 354-364.

- [3] L. Stasiuk, Gender Marked Intimate Conversational Interaction of Spouses in Modern English, CEUR Workshop Proceedings 2870 (2021) 731-742.
- [4] A. Hadzalo, Analysis of Gender-Marked Units: Statistical Approach, CEUR workshop proceedings 2604 (2020) 462-471.
- [5] Y. Butelsky, Statistical Methods to Detect Gender Peculiarities of Communication in Vkontakte Social Network Groups, in Proceedings of the 11th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2016, pp. 132-135. doi: 10.1109/STC-CSIT.2016.7589888.
- [6] I. Afanasieva, N. Golian, V. Golian, A. Khovrat, K. Onyshchenko, Application of Neural Networks to Identify of Fake News, CEUR Workshop Proceedings 3396 (2023) 346-358.
- [7] A. Shupta, O. Barmak, A. Wierzbicki, T. Skrypnyk, An Adaptive Approach to Detecting Fake News Based on Generalized Text Features, CEUR Workshop Proceedings 3387 (2023) 300-310.
- [8] V.-A. Oliinyk, V. Vysotska, Y. Burov, K. Mykich, V. Basto-Fernandes, Propaganda Detection in Text Data Based on NLP and Machine Learning, CEUR workshop proceedings 2631 (2020) 132-144.
- [9] R. A. Dar, Dr. R. Hashmy, A Survey on COVID-19 related Fake News Detection using Machine Learning Models, CEUR Workshop Proceedings 3426 (2023) 36-46.
- [10] V. Vysotska, S. Mazepa, L. Chyrun, O. Brodyak, I. Shakleina, V. Schuchmann, NLP Tool for Extracting Relevant Information from Criminal Reports or Fakes/Propaganda Content, in Proceedings of IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022, pp. 93-98, doi: 10.1109/CSIT56902.2022.10000563.
- [11] A. Mykytiuk, V. Vysotska, O. Markiv, L. Chyrun, Y. Pelekh, Technology of Fake News Recognition Based on Machine Learning Methods, CEUR Workshop Proceedings 3387 (2023) 311-330.
- [12] T. Batiuk, V. Vysotska, V. Lytvyn, Intelligent system for socialization by personal interests on the basis of SEO technologies and methods of machine learning, CEUR workshop proceedings 2604 (2020) 1237-1250.
- [13] D. Uhryn, O. Naum, N. Antonyuk, I. Dyyak, L. Chyrun, A. Demchuk, V. Vysotska, Z. Rybchak, T. Batiuk, Tourist Itineraries Plan Design Based on the Behavior of Bee Colonies, CEUR Workshop Proceedings 2631 (2020) 516-539.
- [14] T. Batiuk, V. Vysotska, R. Holoshchuk, S. Holoshchuk, Intelligent System for Socialization of Individual's with Shared Interests based on NLP, Machine Learning and SEO Technologies, CEUR Workshop Proceedings 3171 (2022) 572-631.
- [15] D. Dosyn, T. Batiuk, A Realization of Visual Biometric Validation to Enhance Guarded and Efficient Authorization for Intellectual Systems, CEUR Workshop Proceedings 3668 (2024) 247-268.
- [16] T. Batiuk, L. Chyrun, O. Oborska, Ontology Model and Ontological Graph for Development of Decision Support System of Personal Socialization by Common Relevant Interests, CEUR Workshop Proceedings 3171 (2022) 877-903.
- [17] R. Bekesh, L. Chyrun, P. Kravets, A. Demchuk, Y. Matseliukh, T. Batiuk, I. Peleshchak, R. Bigun, I. Maiba, Structural modeling of technical text analysis and synthesis processes, CEUR Workshop Proceedings 2604 (2020) 562-589.

- [18] A. Yarovy, D. Kudriavtsev, Method of Multi-Purpose Text Analysis Based on a Combination of Knowledge Bases for Intelligent Chatbot, CEUR Workshop Proceedings 2870, 2021, pp. 1238-1248.
- [19] V. Vasyliuk, Y. Shyika, T. Shestakevych, Information System of Psycholinguistic Text Analysis, CEUR workshop proceedings 2604 (2020) 178-188.
- [20] O. Artemenko, V. Pasichnyk, N. Kunanets, K. Shunevych, Using sentiment text analysis of user reviews in social media for e-tourism mobile recommender systems, CEUR workshop proceedings 2604 (2020) 259-271.
- [21] I. Gruzdo, I. Kyrychenko, G. Tereshchenko, O. Cherednichenko, Application of Paragraphs Vectors Model for Semantic Text Analysis, CEUR workshop proceedings 2604 (2020) 283-293.
- [22] N.B. Shakhovska, R.Yu. Noha, Methods and tools for text analysis of publications to study the functioning of scientific schools, Journal of Automation and Information Sciences 47(12) (2015) 29-43.
- [23] V. Vysotska, V.B. Fernandes, V. Lytvyn, M. Emmerich, M. Hrendus, Method for Determining Linguometric Coefficient Dynamics of Ukrainian Text Content Authorship, Advances in Intelligent Systems and Computing 871 (2019) 132-151. doi: 10.1007/978-3-030-01069-0\_10.
- [24] V. Vysotska, Y. Burov, V. Lytvyn, A. Demchuk, Defining Author's Style for Plagiarism Detection in Academic Environment, in: Proceedings of the International Conference on Data Stream Mining and Processing, DSMP, 2018, pp. 128-133. DOI: 10.1109/DSMP.2018.8478574.
- [25] V. Vysotska, O. Kanishcheva, Y. Hlavcheva, Authorship Identification of the Scientific Text in Ukrainian with Using the Lingvometry Methods, in: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, 2018, pp. 34-38. DOI: 10.1109/STC-CSIT.2018.8526735.
- [26] V. Lytvyn, V. Vysotska, Y. Burov, I. Bobyk, O. Ohirko, The linguometric approach for co-authoring author's style definition, in: International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS-SWS, 2018, pp. 29-34. doi: 10.1109/IDAACS-SWS.2018.8525741.
- [27] V. Lytvyn, V. Vysotska, I. Budz, Y. Pelekh, N. Sokulska, R. Kovalchuk, L. Dzyubyk, O. Tereshchuk, M. Komar, Development of the quantitative method for automated text content authorship attribution based on the statistical analysis of N-grams distribution, Eastern-European Journal of Enterprise Technologies 6(2-102) (2019) 28-51. doi: 10.15587/1729-4061.2019.186834.
- [28] V. Vysotska, O. Markiv, S. Teslia, Y. Romanova, I. Pihulechko, Correlation Analysis of Text Author Identification Results Based on N-Grams Frequency Distribution in Ukrainian Scientific and Technical Articles, CEUR Workshop Proceedings 3171 (2022) 277-314.
- [29] V. Motyka, Y. Stepaniak, M. Nasalska, V. Vysotska, Lexical Diversity Parameters Analysis for Author's Styles in Scientific and Technical Publications, CEUR Workshop Proceedings 3403 (2023) 595-617.
- [30] R. Romanchuk, V. Vysotska, V. Andrunyk, L. Chyrun, S. Chyrun, O. Brodyak, Intellectual Analysis System Project for Ukrainian-language Artistic Works to

Determine the Text Authorship Attribution Probability, in Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT, Lviv, 19-21 October 2023 p.

- [31] O. Levchenko, M. Dilai, Qualitative and Quantitative Markers of Individual Authorial Conceptualization, CEUR Workshop Proceedings 3396 (2023) 1-19.
- [32] I. Khomytska, V. Teslyuk, I. Bazylevych, I. Karamysheva, Automated Identification of Authorial Styles, CEUR Workshop Proceedings 3396 (2023) 323-333.
- [33] I. Butko, The use of geospatial information by public authorities to support the decision making of management. *Advanced Information Systems* 5(1) (2021) 39–44. doi: 10.20998/2522-9052.2021.1.05.
- [34] V. Shynkarenko, I. Demidovich, Natural Language Texts Authorship Establishing Based on the Sentences Structure, CEUR Workshop Proceedings 3171 (2022) 328-337.
- [35] Y. Hlavcheva, O. Kanishcheva, M. Vovk, M. Glavchev, Identification of the Author's Idea Based on the Modified TextRank Method, CEUR Workshop Proceedings 2870 (2021) 118-128.
- [36] V. Shynkarenko, I. Demidovich, Authorship Determination of Natural Language Texts by Several Classes of Indicators with Customizable Weights, CEUR Workshop Proceedings 2870 (2021) 832-844.
- [37] I. Khomytska, V. Teslyuk, The Multifactor Method Applied for Authorship Attribution on the Phonological Level, CEUR workshop proceedings 2604 (2020) 189-198.
- [38] I. Khomytska, V. Teslyuk, A. Holovatyy, O. Morushko, Development of methods, models, and means for the author attribution of a text, *Eastern-European Journal of Enterprise Technologies*. 3(2(93)) (2018) 41–46. doi: 10.15587/1729-4061.2018.132052.
- [39] I. Khomytska, V. Teslyuk, Authorship and Style Attribution by Statistical Methods of Style Differentiation on the Phonological Level, *Advances in Intelligent Systems and Computing* 871 (2019) 105–118. doi: 10.1007/978-3-030-01069-0\_8.
- [40] Y. Zhao, J. Da, J. Yan, Detecting health misinformation in online health communities: Incorporating behavioral features into machine learning based approaches, *Information Processing & Management* 58(1) (2021) 102390.
- [41] M. Hartmann, Y. Golovchenko, I. Augenstein, Mapping (dis-)information flow about the MH17 plane crash, arXiv:1910.01363, 2019.
- [42] S. Ahmed, Classification of Censored Tweets in Chinese Language using XLNet, in Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, 2021, pp. 136-139.
- [43] V. Vysotska Modern state and prospects of information technologies development for natural language content processing, CEUR Workshop Proceedings 3668 (2024) 198–234.
- [44] I. Zamaruieva, S. Lienkov, O. Babich, A. Shevchenko, Y. Khlaponin, N. Bernaz, Analytical Approaches to News Content Processing during the War in Ukraine in Opposing Geopolitical Alliances Mass Media, CEUR Workshop Proceedings 3403 (2023) 618-631.



- [45] V. Vysotska, Computer Linguistic Systems Design and Development Features for Ukrainian Language Content Processing, CEUR Workshop Proceedings 3688 (2024) 229–271. URL: <https://ceur-ws.org/Vol-3688/paper18.pdf>.
- [46] S. Albota, Creating a Model of War and Pandemic Apprehension: Textual Semantic Analysis, CEUR Workshop Proceedings 3396 (2023) 228-243.
- [47] N. Khairova, Y. Holyk, D. Sytnikov, Y. Mishcheriakov, N. Shanidze, Topic Modelling of Ukraine War-Related News Using Latent Dirichlet Allocation with Collapsed Gibbs Sampling, CEUR Workshop Proceedings 3688 (2024) 1-15.
- [48] S. Mainych, A. Bulhakova, V. Vysotska, Cluster Analysis of Discussions Change Dynamics on Twitter about War in Ukraine, CEUR Workshop Proceedings 3396 (2023) 490-530.
- [49] R. Nazarchuk, S. Albota, Tweets about Ukraine during the russian-Ukrainian War: Quantitative Characteristics and Sentiment Analysis, CEUR Workshop Proceedings 3426 (2023) 551-560.
- [50] N. Khairova, A. Kolesnyk, O. Mamyrbayev, K. Mukhsina, The Aligned Kazakh-Russian Parallel Corpus Focused on the Criminal Theme, CEUR Workshop Proceedings 2362 (2019) 116-125.
- [51] S. Voloshyn, V. Vysotska, O. Markiv, I. Dyyak, I. Budz, V. Schuchmann, Sentiment Analysis Technology of English Newspapers Quotes Based on Neural Network as Public Opinion Influences Identification Tool, in Proceedings of 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022, pp. 83-88, doi: 10.1109/CSIT56902.2022.10000627.
- [52] N. Khairova, A. Shapovalova, O. Mamyrbayev, N. Sharonova, K. Mukhsina, Using BERT model to Identify Sentences Paraphrase in the News Corpus, CEUR Workshop Proceedings 3171 (2022) 38-48.
- [53] N. Bondarchuk, I. Bekhta, O. Melnychuk, O. Matviienkiv, Keyword-based Study of Thematic Vocabulary in British Weather News, CEUR Workshop Proceedings 3171 (2022) 451-460.
- [54] S. Voloshyn, O. Markiv, V. Vysotska, I. Dyyak, L. Chyrun, V. Panasyuk, Emotion Recognition System Project of English Newspapers to Regional E-Business Adaptation, Proceedings of IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022, pp. 392-397, doi: 10.1109/CSIT56902.2022.10000527.
- [55] N. Antonyuk, L. Chyrun, V. Andrunyk, A. Vasevych, S. Chyrun, A. Gozhyj, I. Kalinina, Y. Borzov, Medical news aggregation and ranking of taking into account the user needs, CEUR Workshop Proceedingsn248 (2019) 369–382.
- [56] V. Andrunyk, A. Vasevych, L. Chyrun, N. Chernovol, N. Antonyuk, A. Gozhyj, V. Gozhyj, I. Kalinina, M. Korobchynskiy, Development of information system for aggregation and ranking of news taking into account the user needs, CEUR Workshop Proceedings 2604 (2020) 1127–1171.
- [57] V. Vysotska, S. Voloshyn, O. Markiv, O. Brodyak, N. Sokulska, V. Panasyuk, Tone Analysis of Regional Articles in English-Language Newspapers Based on Recurrent Neural Network Bi-LSTM, in Proceedings of the 5th International Conference on Advanced Information and Communication Technologies (AICT), 2023, pp. 158-163.

- [58] S. Albota, Linguistic and Psychological Features of the Reddit News Post, in Proceedings of the IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2020, 1, pp. 295–299.
- [59] N. Shakhovska, M. Medykovskij, L. Bychkovska, Building a smart news annotation system for further evaluation of news validity and reliability of their sources, *Przeglad Elektrotechniczny* 91(7) (2015) 43-44.
- [60] V. Vysotska, R. Holoshchuk, S. Goloshchuk, O. Voloshynskiy, M. Shevchenko, V. Panasyuk, Predicting the Effects of News on the Financial Market Based on Machine Learning Technology, in Proceedings of the 5th International Conference on Advanced Information and Communication Technologies (AICT), 2023, pp. 152-157.
- [61] Chew, R., Kery, C., Baum, L., Bukowski, T., Kim, A., & Navarro, M. (2021). Predicting age groups of Reddit users based on posting behavior and metadata: classification model development and validation. *JMIR Public Health and Surveillance*, 7(3), e25807.
- [62] Z. Miller, B. Dickinson, W. Hu, Gender Prediction on Twitter Using Stream Algorithms with N-Gram Character Features, *International Journal of Intelligence Science* 2 (4A) (2012) 24184, doi:10.4236/ijis.2012.224019.
- [63] S. Rosenthal, K. McKeown, Age prediction in blogs: A study of style, content, and online behavior in pre-and post-social media generations, in: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies. 2011, pp. 763-772.
- [64] D. Nguyen, N. A. Smith, C. P. Ros'e, Author age prediction from text using linear regression, in: Proceedings of the 5th ACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, LaTeCH@ ACL 2011, 24 June, 2011, Portland, Oregon, USA. Association for Computational Linguistics, 2011, pp. 115-123.
- [65] D. Nguyen, D. Trieschnigg, A. S. Dogruoz", R. Grave, M. Theune, T. Meder, F. de Jong, Why gender and age prediction from tweets is hard: Lessons from a crowdsourcing experiment, in: Proceedings of the Technical Papers 25th International Conference on Computational Linguistics, August 23-29, 2014, Dublin, Ireland. Association for Computational Linguistics, 2014, pp. 1950-1961.
- [66] I. Khomytska, V. Teslyuk, I. Bazylevych, I. Shylinska, Approach for minimization of phoneme groups in authorship attribution, *International Journal of Computing* 19(1) (2020) 55-62.
- [67] I. Khomytska, V. Teslyuk, A. Holovatyy, O. Morushko, Development of Methods, Models and Means for the Author Attribution of a Text, *Eastern-European Journal of Enterprise Technologies* 3/2 (93) (2018) 41–46.
- [68] I. Khomytska, V. Teslyuk, Authorship Attribution by Differentiation of Phonostatistical Structures of Styles, in: Proceedings of the XIIIth Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, Lviv, 2018, pp. 5–8.
- [69] I. Khomytska, V. Teslyuk, The Software for Authorship and Style Attribution, in: Proceedings of the 15th International Conference on CADMS, Polyana, 2019, pp. 23–26.

- [70] I. Khomytska, V. Teslyuk, Mathematical Methods Applied for Authorship Attribution on the Phonological Level, in: Proceedings of the XIVth Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, Lviv, 2019, pp. 7–11.
- [71] I. Khomytska, V. Teslyuk, L. Bordyuk, The Kolmogorov-Smirnov's Test for Authorship Attribution on the Phonological Level, in: Proceedings of the IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 2020, pp. 259–262.
- [72] I. Khomytska, V. Teslyuk, N. Kryvinska, V. Beregovskiy, The nonparametric method for differentiation of phonostatistical structures of authorial style, *Procedia Computer Science* 160 (2019) 38–45.
- [73] Dataset of Blog Authorship Corpus. URL: <https://www.kaggle.com/datasets/rtatman/blog-authorship-corpus>.
- [74] Dataset of Spooky Author Identification. URL: <https://www.kaggle.com/competitions/spooky-author-identification/data?select=train.zip>.
- [75] O. Prokipchuk, V. Vysotska, P. Pukach, V. Lytvyn, D. Uhryn, Y. Ushenko, Z. Hu, Intelligent Analysis of Ukrainian-language Tweets for Public Opinion Research based on NLP Methods and Machine Learning Technology, *International Journal of Modern Education and Computer Science* 15(3) (2023) 70–93.