Technology of Fake News Recognition Based on Machine Learning Methods

Andrii Mykytiuk [1], Victoria Vysotska [1,2], Oksana Markiv [1], Lyubomyr Chyrun [3] and Yaroslav Pelekh [1]

[1] *Lviv Polytechnic National University, S. Bandera Street, 12, Lviv, 79013, Ukraine*
[2] *Osnabrück University, Friedrich-Janssen-Str. 1, Osnabrück, 49076, Germany*
[3] *Ivan Franko National University of Lviv, University Street, 1, Lviv, 79000, Ukraine*

Abstract

The article dwells upon the system for detecting fake news via the Internet based on machine learning methods. In the process of analyzing the accuracy of fakes or non-fakes identification, several algorithms have been chosen using the multilayer parceptron (0.9945/0.9979), the Bayes classifier (0.913/0.998), the random forest (0.933/0.991), the logistic regression (0.9988/0.9965), the k-nearest neighbors (0.999/0.83) and the decision tree (0.9903/0.98) to select the most optimal and accurate one according to the results of the experimental trials. It is highlighted that despite the method of k-nearest neighbors has shown the best results (0.999), it has the worst result when recognizing non-fakes (0.83). It is also proved that in order to identify propaganda and fakes, the best variant is to use the method of k-nearest neighbors (0.999), the logistic regression (0.9988) and the multilayer parceptron (0.9945). The Bayesian classifier (0.998), the multilayer parceptron (0.9979) and the logistic regression (0.9965) are more optimal for identifying real news (not fake news). For the simultaneous recognition of fakes and real news that are not fakes, when analyzing the differences in identification accuracy, the multilayer parceptron (0.0034) and the logistic regression (0.0023) are the best ones. Other algorithms have the following worse results: the decision tree (0.013), the random forest (0.058), the Bayes classifier (0.085) and the k-nearest neighbors (0.169).

Keywords

News, fake, propaganda, fake recognition, machine learning, k-nearest neighbors, decision tree, logistic regression, random forest, Bayes classifier, multilayer parceptron

1. Introduction

Fake news is false or misleading information often designed to damage the reputation of an individual or legal entity or to earn advertising revenue [1]. In addition, disinformation is an insidious type of malicious misinformation that is sometimes created and disseminated by hostile foreign representatives, especially during elections or wars [2]. Some definitions of fake news include satirical articles that are misinterpreted as genuine, and articles that use sensationalism or bait that is not supported by the text. About 45% of adults believe that they encounter fakes every day and this number is growing [1-3]. The deterioration of the situation in the world has also contributed to the increase in the amount of misinformation. People watch dozens or even hundreds of news stories every day, and this can cause significant damage to society as the number of fakes is increasing [4-5].

The problem is getting worse every day despite governments around the world are trying to reduce the spread of fake news through social networking platforms. With the transition to online format,

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: andrii.mykytiuk.msaad.2022@lpnu.ua (A. Mykytiuk); victoria.a.vysotska@lpnu.ua (V. Vysotska); oksana.o.markiv@lpnu.ua (O. Markiv); Lyubomyr.Chyrun@lnu.edu.ua (L. Chyrun); yaroslav.m.pelekh@lpnu.ua (Y. Pelekh)
ORCID: 0009-0003-9906-6156 (A. Mykytiuk); 0000-0001-6417-3689 (V. Vysotska); 0000-0002- 1691-1357 (O. Markiv); 0000- 0002-9448-1751 (L. Chyrun); 0000-0002-4339-8093 (Y. Pelekh)

programs based on machine learning can help people to distinguish fakes [6-9]. But it is still difficult to classify news in social networks according to their characteristics [10-15].

The goal of the research is to create a system that can recognize fake news via the Internet. So, it is possible to distinguish, the following specific tasks:

1. Study scientific and research literature, which examines various methods of combating fakes.
2. Compare existing analog systems.
3. Describe the functionality of the software system.
4. Analyze the requirements and describe the functions of the system and present them accordingly using the IBM Rational Rose software tool.
5. Choose the software solution and implement the software tool.

The object of the study is a large number of fakes, which creates many problems for an ordinary user. The subject of research is the system of recognition of fake news.

2. Related works

The well-known American broadcaster NBC News says that the lie is spreading much faster than the truth. Meanwhile, German company M/s "Statista" states that fake news is an insidious and widespread problem in the news industry in general. The most recent example of this is the COVID-19 pandemic, when nearly 80 percent of consumers in the United States have faced with fake news about the coronavirus outbreak. For example, the spring survey (2020) has shown that 60 percent of young people in the United Kingdom have recently used social media for information about the coronavirus, and 59 percent have faced with fake news concerning the theme. Moreover, about 52 percent of Americans say that they regularly encounter fake news online.

Moreover, in France, nearly 30 percent of young people have used social media as their main source of information about the coronavirus. Consciously or unconsciously, many consumers observe fake news and share it. Unfortunately, it is a time-consuming process to eliminate fake news, so regular news consumers need to develop awareness and improve their ability to identify false information [16-21].

For example, The Reuters Institute highlights that 41 percent of Americans actively avoid watching or reading the news because of false information saying that reading the news makes them feel sad and depressed about the current state of the world. In addition, the Pew Research Center states that 53 percent of the Americans get their news from social media, according to a 2020 survey. Unfortunately, not many users are in the habit of making an active effort to reduce polarizing content [22-34].

The NBC News reports that during the second quarter of 2020, Facebook has removed seven million posts containing fake news, false or unverified information about the ongoing COVID-19 pandemic, so such removal was also a way to prevent this false information from endangering people's health [35-45]. Moreover, the German Marshall Fund says that there were 1.8 billion fake news posts on Facebook during the third quarter of 2020. According to statistics, this is 242 percent more than the 500 million cases before the 2016 election. Studying information about the topic of work, seven basic analogue decisions on the market have been distinguished:

1. Bot Sentinel that is designed for detecting and monitoring unreliable tollbooths and accounts in Twitter and detect such bots examining their impact on discourse (Fig. 1). Advantages: free of charge, can recognize photos, available on the most popular platforms. Disadvantages: it works only with English, does not support the Opera browser, works only with Twitter.
2. CaptainFact that checks web-content based on browser extension providing video overlays with source and contextual data (Fig. 2). Advantages: it is distributed as a free browser extension, checks videos, has a discussion platform. Disadvantages: only video works at the moment.
3. ClaimBuster that is the web-based, real-time tool based on natural language processing and supervised learning (based on a human-encoded dataset) to detect factual and false information (Fig. 3). Advantages: free, wide space for application, open API. Disadvantages: only supports English.
4. FakerFact that evaluates the purpose and characteristics of information, does not rate an article as true or false, but rather assesses its purpose and objectivity (Fig. 4). Advantages: many evaluation criteria. Disadvantages: does not show whether it is fake or not.

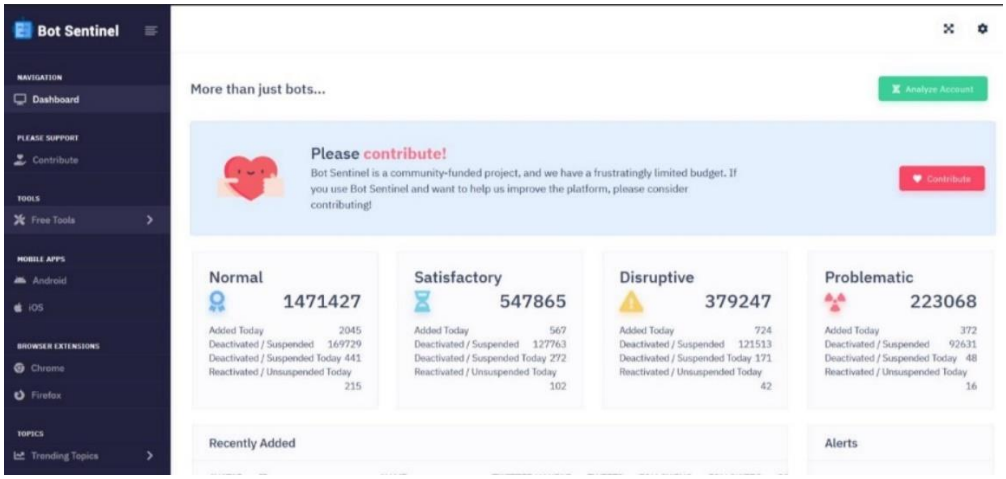

Figure 1: Bot Sentinel

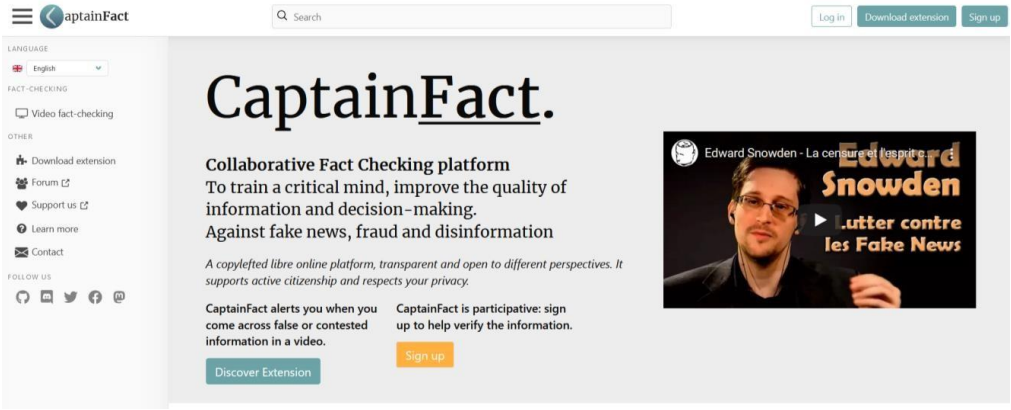

Figure 2: CaptainFact

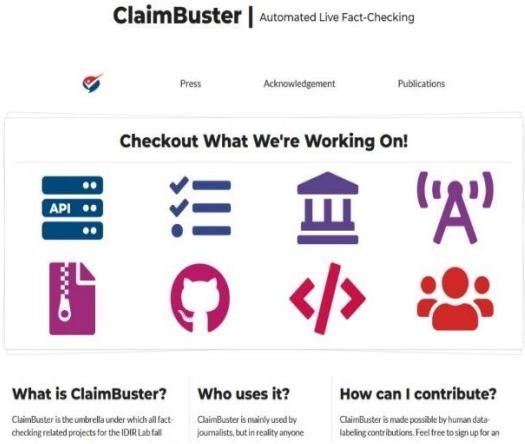

Figure 3: ClaimBuster

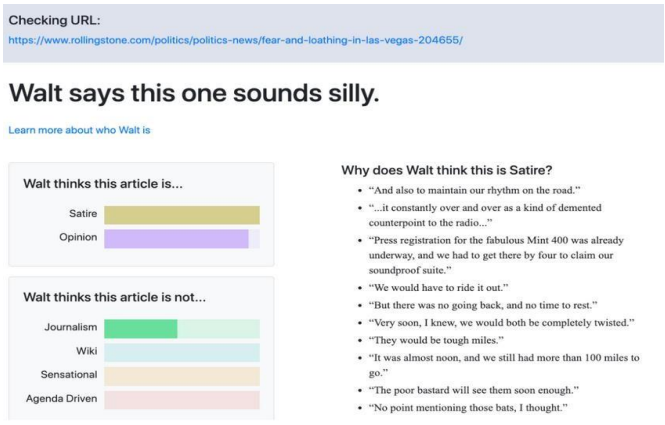

Figure 4: FakerFact

5. Hxy visualizes the distribution of articles on the Internet, calculating a bot score, that is a measure of the likely level of automation and is good for studying how social media affects public discourse (Fig. 5). Advantages: large number of languages, good visualization, convenient tools. Cons: beta version, only works with twitter.
6. NewsCheck that is a web platform that assesses credibility using a combination of machine technology (blockchain) and humans to combat fake news, identifies biases (Fig. 6). Advantages: it works using blockchain, they have a content management and delivery system, there is a set of journalistic standards. Disadvantages: works only with English.
7. Our.news that is a website, browser extension and app that provides crowdsourced fact-checking (Fig. 7). In addition, bias detection algorithms are used to evaluate user ratings. Fact-checking information includes bias information, sources, and links to Politifact and Snopes when

appropriate. Advantages: it has many different evaluation criteria. Disadvantages: it does not use machine learning.

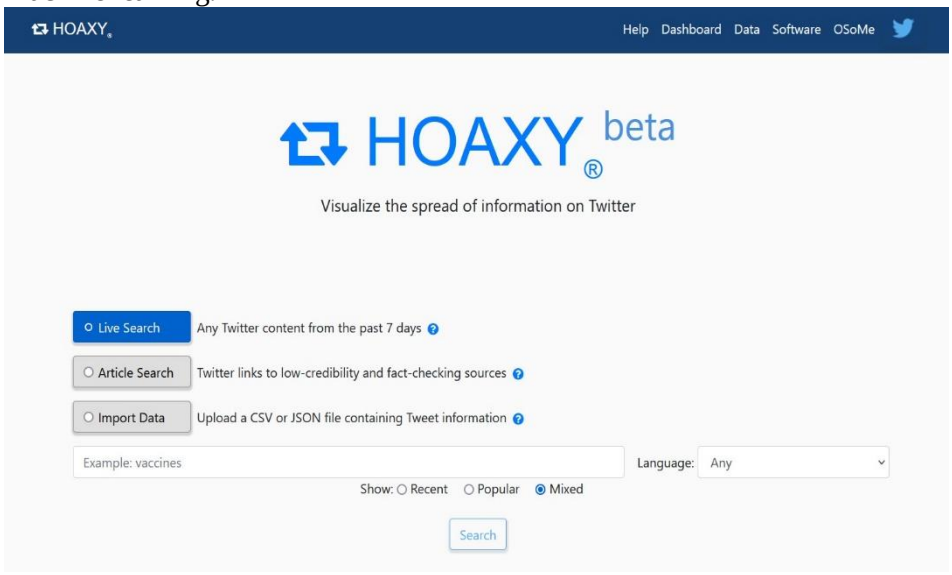

Figure 5: Hoaxy

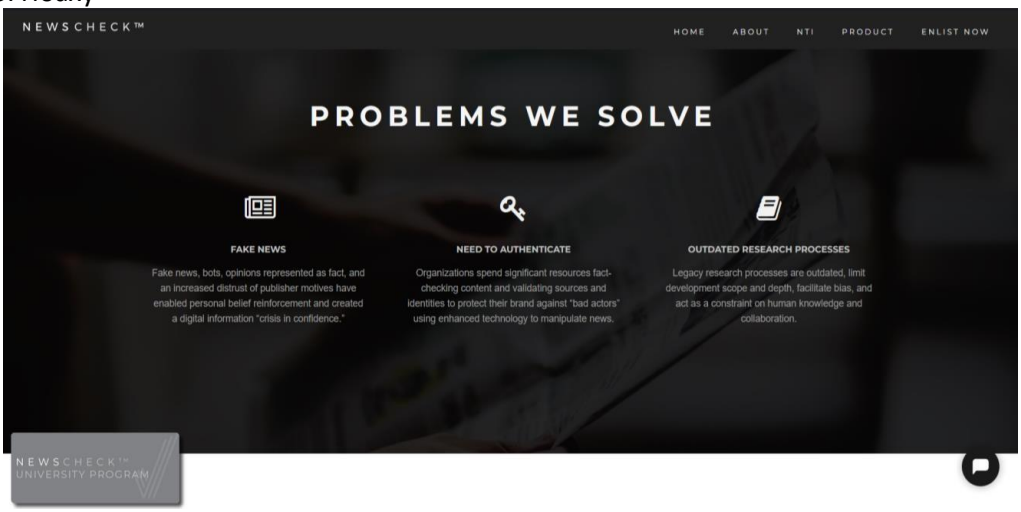

Figure 6: NewsCheck

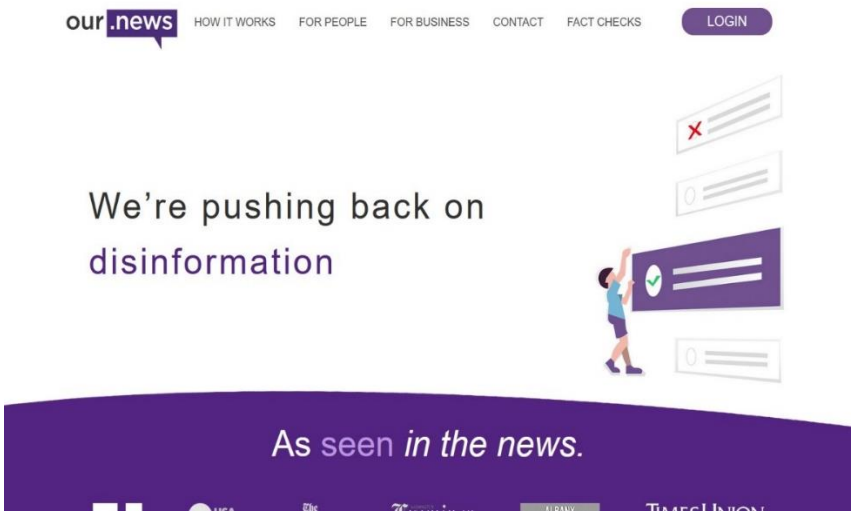

Figure 7: Cactus Spam Filter

For evaluation, the evaluation of advantages [46-51]: better/worse has been chosen, and the rating scale has been developed.

Localization:

- Bad – only one language; *
- Good – 5 languages; **
- Super – 10+ languages; ***

Browser:

- Bad – 1; *
- Good – 2; **
- Super – 3; ***

Mail clients:

- Bad – 1; *
- Good – 3; **
- Super – 5+; ***

Payment:

- Bad - subscription; *
- Good - purchase; **
- Super - free; ***

Flexibility:

- Bad – there are no settings for filters and lists; *
- Good – limited number of settings; **
- Super – the ability to configure anything; ***

Self-study:

- Bad - none; *
- Super - there is. ***

The comparison table for the project and analogues has been constructed:

Table 1

Set of characteristics and criteria for evaluation and comparison

Characteristics	Project	1	2	3	4	5	6	7
Localization	***	***	***	**	*	**	*	*
Browser	***	**	**	**	**	**	**	**
Mail clients	**	*	**	***	**	**	***	***
Payment	***	**	***	**	**	**	***	***
Flexibility	***	**	***	**	**	**	***	***
Self-study	***	*	*	***	***	*	*	***

3. Methods

When choosing the algorithm for work, seven algorithms (with the help of the multilayered parceptron, Bayes classifier, the random forest, the logistical regression, the k- closest neighbors and the tree of decisions) have been selected. And the most suitable for the task should be chosen in experimental way [6-9].

1. **The method of k nearest neighbors.** In order to make a prediction for a new data point, the algorithm finds the point in the training set that is the closest to the new point. Then it assigns the label belonging to training set point to the new data point. The k in the k-nearest neighbors method means that instead of using only the nearest neighbor of a new data point, it is possible to consider any fixed number (k) of neighbors during training (for example, consider the nearest three or five neighbors) and then make a prediction for a data point using the class to which most of its neighbors belong. This is the simplest machine learning algorithm. So, the question is how to choose the value of *k*? Choosing the right *k* value is called parameter tuning and is necessary for better results. By choosing the value of *k*, it is possible to calculate the square root of the total number of data points available in the data set:

k=N, where N is the total number of data. Also, to avoid confusion between the 2 classes, an odd value of *k* is always chosen. The KNN algorithm is used when the data is correctly labeled. For example, if predicting whether a letter is fake or not, then the final label can be 1 or 0. It cannot be a number or -1. It is also recommended that the data set is not very large.

2. **Linear regression**. This method finds the parameters w and b that minimize the mean squared error between the predicted and actual responses "*y*" in the training set. The root mean square error is equal to the sum of the squares of the differences between the predicted and actual values. In addition, linear regression was the first type of regression analysis that was thoroughly studied and began to be widely used in practical applications. This is due to the fact that it is easier to estimate parameters in linear models, and also to the fact that the statistical properties of the obtained estimates are easier to determine.

Scikit-learn realizes three types of naive Bayesian classifiers: GaussianNB, BernoulliNB and MultinomialNB. GaussianNB can be applied to any continuous data, while BernoulliNB accepts the binary data, MultinomialNB accepts the discrete data (i.e., each feature is a count of integer values of a particular feature). BernoulliNB and MultinomialNB are mainly used to classify text data.

If the dependence between one input and one output variable is considered, then the simple linear regression takes place. For this, the regression equation $y = ax + b$ is determined and the corresponding straight line, known as the regression line, is constructed. Coefficients *a* and *b*, which are also called parameters of the model, are determined in such a way that the sum of the squared deviations of the points corresponding to the real observations of the data from the regression line would be minimal. The coefficients are usually estimated by the method of least squares.

If the relationship between several input and one output variable is sought, then multiple linear regression takes place. The corresponding equation has the form:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n, \quad (1)$$

where *n* is the number of input variables. It is obvious that in this case the model will not be described by a straight line, but by a hyperplane. The coefficients of the multiple linear regression equation are chosen to minimize the sum of the squares of the deviations of the real data points from this hyperplane.

3. **Random forest method.** It is a method that works by building multiple decision trees during model training and produces a fashion for classes (classifications) or an averaged prediction (regression) of the trees built. However, its big drawback is the tendency to re-learn. Individual decision trees are generated using attribute selection metrics such as information gain, gain, and Gini index for each attribute. Each tree depends on an independent random sample. In the classification problem, each tree votes, and the final result is the most popular class. In the case of regression, the average value of all outputs of the tree is considered the final result. The algorithm is simpler and more powerful compared to other nonlinear classification algorithms. In general, the algorithm works in four steps:

1. The random samples are selected from a given data set.
2. The decision trees are constructed for each sample and the prediction result from each decision tree is obtained.
3. The vote is held for each predicted outcome.
4. The result of the forecast with the highest number of votes is chosen as the final forecast.

Random forests are considered a highly accurate and reliable method due to the number of decision trees involved in the process. Random forests generate predictions slowly because they have multiple decision trees. Each time it makes a prediction, all the trees in the forest must make a prediction for the same data and then vote on it. This whole process is time-consuming.

4. **Logistic regression** is a statistical regression method used when the dependent variable is categorical, that is, it can take on only two values (or, more generally, a finite set of values). This algorithm uses non-linear functions, which are also called logistic functions or sigmoid functions. A sigmoid function (logistic function) is a function that resembles an S-shaped curve when plotted. It takes values between 0 and 1 and "squeezes" them to the upper and lower bounds, marking them as 0 or 1. The equation for the sigmoid function is: $y=1/(1+e^{-x})$, where *e* is an exponential function or an exponential constant, and has the value approximately 2.71828. This gives a value of *y* that is extremely close to 0 if *x* is a large negative value and close to 1; if *x* is a large positive value. Once the input values have been compressed to 0 or 1, they can be passed through a typical linear function, but can now be assigned to different categories.

When using logistic regression, a threshold is usually specified that indicates at what value an example will be placed in one class compared to another class. In a fake classification task, a threshold of 0.5 could be set, which would cause an article with a 50% or greater probability of being fake to be classified as "fake", and any article with a probability of less than 50% to be classified as "authentic".

5. **Method of support vectors**. In this method, part of the points of the training set is important for determining the decision-making boundary: the points that lie on the border between the classes. They are called support vectors and gave their name to the support vector machine. To obtain a prediction for a new point, the distance to each reference vector is measured. The classification decision is made based on the distances to the support vectors, as well as the importance of the support vectors obtained in the training process (stored in the dual_coef_ attribute of the SVC class). The distance between data points is measured using a Gaussian kernel. The advantages of this method are that it works well on different data sets, allows to build complex boundaries even if there are only a few features, and works well with data of different volumes.

6. **Multilayer parceptron** is a special case of a Rosenblatt perceptron in which a single error backpropagation algorithm trains all layers. A feature is the presence of more than one pupil of the layer (as a rule, two or three, there is currently no justification for using a larger number - speed is lost without acquiring quality). The need for a large number of learner layers is eliminated, since theoretically a single hidden layer is sufficient to transcode the input signal in such a way as to obtain a linear map for the output signal. But there is an assumption that by using a larger number of layers, it is possible to reduce the number of elements in them, that is, the total number of elements in the layers will be less than when using one hidden layer.

MLPs have shown the ability to find approximate solutions for extremely complex problems. In particular, they are a universal approximator of functions, therefore they are successfully used in the construction of regression models. Since classification can be considered as a special case of regression when the output variable is categorical, classifiers can be built based on MLP.

For the first time, the multilayer perceptron was proposed by F. Rosenlatt. However, in the form in which it is currently used, the multilayer perceptron was developed by D. Rumelhart.

Rumelhart's perceptron differs from Rosenblatt's perceptron in the following properties:

- use of a non-linear activation function;
- the number of hidden layers is more than one (usually no more than three); input signals are binary, and coded with decimal numbers normalized to interval [0,1]; the original error of the network is not defined as the number of false recognitions
- examples, but as a certain meaning of the inconsistency; training is not carried out to minimize the error, but to stabilize the weights of the network, which avoids retraining.

All these methods are related to machine learning with a teacher and the input of the algorithm is object-response pairs, and the algorithm finds a way to get an answer by object. The two main tasks of such training are classification and regression.

4. Experiments

The Python programming language and the Anaconda distribution have been chosen to work on the system.

When choosing a language, it came down to two options; Phyton and R. Both languages are well suited for working with machine learning and data analysis tasks due to their extensive toolkit. But after their detailed analysis and comparison, it was decided to choose Phyton. It has a simple and clear syntax, and many ready-made libraries that will greatly help in development. Also, Python has excellent, simple and clear documentation, and thanks to its popularity and prevalence, the answers to most questions that may arise will most likely be quite easy to find on resources such as Stackoverflow and others.

Anaconda has been chosen for its versatility. It includes everything that is needed for work starting with the convenient Spyder IDE, which includes a code editor and flexible interface and a good set of built-in libraries, and ending with the built-in QT editor for creating graphical interfaces.

The following libraries have been used during the work: pandas, sklearn, numpy, seaborn, matplotlib and pyqt. With the help of pandas, sklearn and numpy, processing and work with the data has been carried out, and with the help of seaborn and matplotlib, the visualization of the obtained results has

been done, and also with the help of graphs. Each of these libraries has been selected as the most suitable in its domain, as well as for having simple and clear documentation.

All the selected toolkit is actually a standard for any work that includes machine learning and data processing. Without them, it would be very difficult to implement any of this.

When creating the program, a dataset with almost 21,000 articles has been used.

- https://www.kaggle.com/datasets/algord/fake-news
- https://www.kaggle.com/code/coshgunrahimli/fake-news-90-accuracy
- https://www.kaggle.com/code/mayankjagtap/fake-news-analysis-by-mayank
- https://www.kaggle.com/code/kkhandekar/real-or-fake-simple-bidirectional-rnn

All of them were publicly available and taken from the Kaggle resource, where it is possible to find many datasets on different topics in different formats and sizes.

Description of the expanded case scenario according to the RUP standard:

1) The stakeholders of the precedent and their requirements:

- The user: wants to receive only true information.
- The software developer: wants feedback to further improve their product.

2) The user of the software system:

The person who wants to save time and not fall into the trap of disinformation with the help of the software system to be developed.

3) The prerequisites of the precedent:

- The user has read and accepted the license terms.
- The user has installed and configured the necessary software on the device.
- The user has given the program access to his browser.

4) The main successful scenario:

- The User runs the Software.
- After that, when viewing the news, the system should inform about the reliability of the information.

5) The alternative scenario:

A) The user does not agree with the license terms.

The user goes in search of other Software.

B) The software failure.

- The user contacts technical support or the developer.
- The developer fixes the bug and provides a patched version or provides instructions on how to fix the bug.

6) The post-conditions:

- The user can check for himself whether the news is authentic and mark it in the program as fake or not.
- The user can create a report with statistics.
- The user can improve the performance of the system by checking.
- The user leaves feedback on the used Software.
- The obtained results are entered into the database.

7) The special system requirements:

- It is necessary to ensure 100% accuracy in recognizing fakes.
- It is necessary to ensure the possibility of localization of the user interface. It is necessary to ensure the reliability and uninterrupted operation of the software.

8) The list of necessary technologies and additional devices:

The User must have a device with an Internet connection to be able to download and install the necessary Software.

To build a diagram of the use cases for the fake article recognition system (Fig. 8), it is firstly necessary to define the main actors: User and Administrator.

The user is associated with the usage options: "Install software" which includes "Register", "Install software" and "Perform initial settings"; "Determine the veracity of the news" in turn is related to "Check the source", "Check the correspondence of the title to the content", "Check the originality of the media", "Check the content of the article", "Complain about the article", "Approve the article".

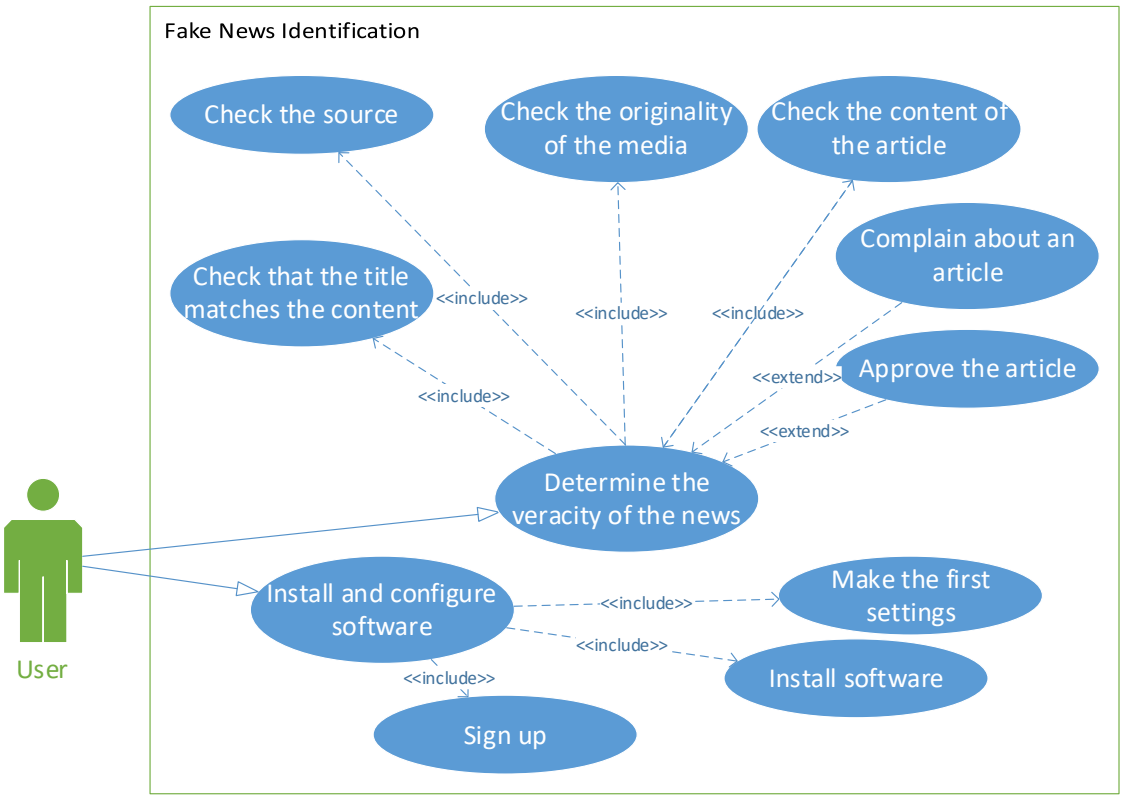

Figure 8: The use case diagram for the fake news system

When constructing the class diagram for a fake news detection system, five main classes have been identified: Reader, News Site, Administrator, Database, System (Fig. 9). One attribute with the visibility quantifier public (name: str) and three operations with the visibility quantifier public (run the program, set parameters, write feedback) have been defined for the Reader class. For the Administrator class, one attribute has been defined with the vAsibility quantifier private (name: str) and three operations with the visibility quantifier public (start the program, set parameters, make changes to the program).

For the System class, three attributes with the visibility quantifier public (name: str, size: int, version: int) and two operations with the visibility quantifier public (evaluate veracity, warn about falsity) have been defined, which will work with the attributes of the News Site class.

For the Database class, two attributes with visibility quantifier public (version: int, content: str) and one operation with visibility quantifier public (to be updated) have been defined.

One attribute with a visibility quantifier has been defined for the News Site class

public (author: str, title: str, content: str, media)

All classes have been connected by an association relation and have corresponding multiplicities.

Six anonymous objects and eleven messages have been determined to construct a sequence diagram for the spam filtering system (Fig. 10). The object formed by the User class has five simple messages, two of which go to the object formed by the System class (run the software, configure) and one message to the object formed by the Administrator class (provide feedback on the program's operation) and the message to the object of the Site with news class (evaluate the site and open the site). The object created by the System class has three simple messages, one to the object created by the Database class (compare with the database), one to the object created by the Site class (scan the site), and a message to the Reader (show the result). One simple message goes from the object created by the Administrator class to the object created by the software class (update the program).

The following activities and transitions have been defined for the construction of the activity diagram for the system (Fig. 11):

1) Add the initial and final state to the diagram.

2) Add the following activities: Run the program, Evaluate reliability, File a complaint, Continue viewing the article, Leave the site, End work.

3) Add two solution symbols.

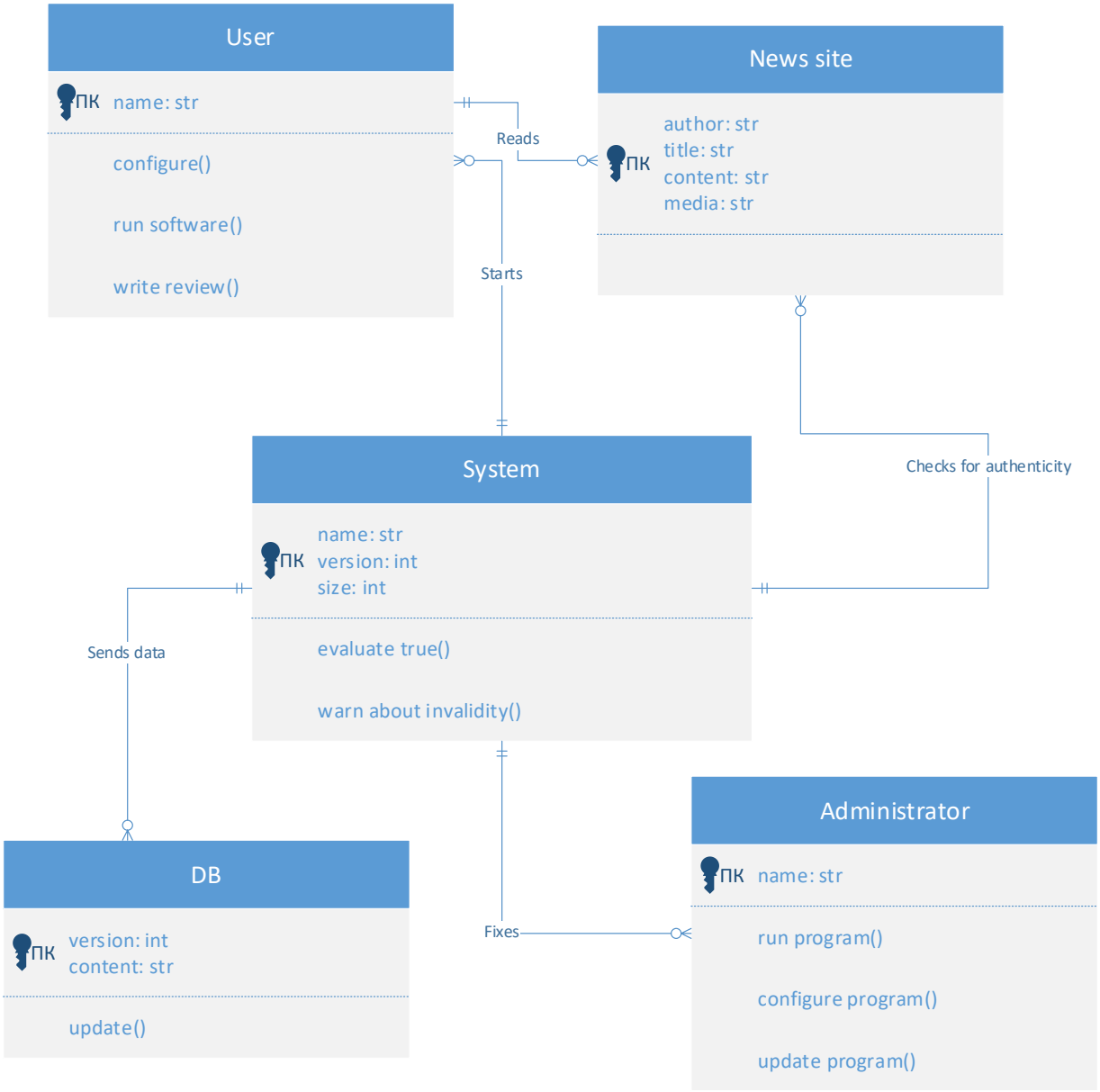

Figure 9: The class diagram for the fake recognition system.

4) Add the routed transition from the initial state to the Run program activity.

5) Add the routed transition from the Run the program activity to the Evaluate reliability activity.

6) Add the transition directed from the activity Evaluate reliability to the separation symbol.

7) Add the transition with guard condition article reliable separation to the activity Continue browsing.

8) Add the transition with the guard condition article is unreliable from the separation symbol to the activity File a complaint.

9) Add the transition from the Continue browsing activity to the connection symbol.

10) Add the transition from the connection symbol to the Leave site activity.

11) Add the routed transition from the activity Leave the site to the activity End work.

12) Add the transition from the Terminate the program activity to the final state.

To construct the state diagram, the following states and transitions should be identified (Fig. 12):

- Add states: Starting the program, Analyzing the article, Complaint, Reading, Completing the program.
- Add the final and initial states.
- Add the transition from the initial state to the program launch state.

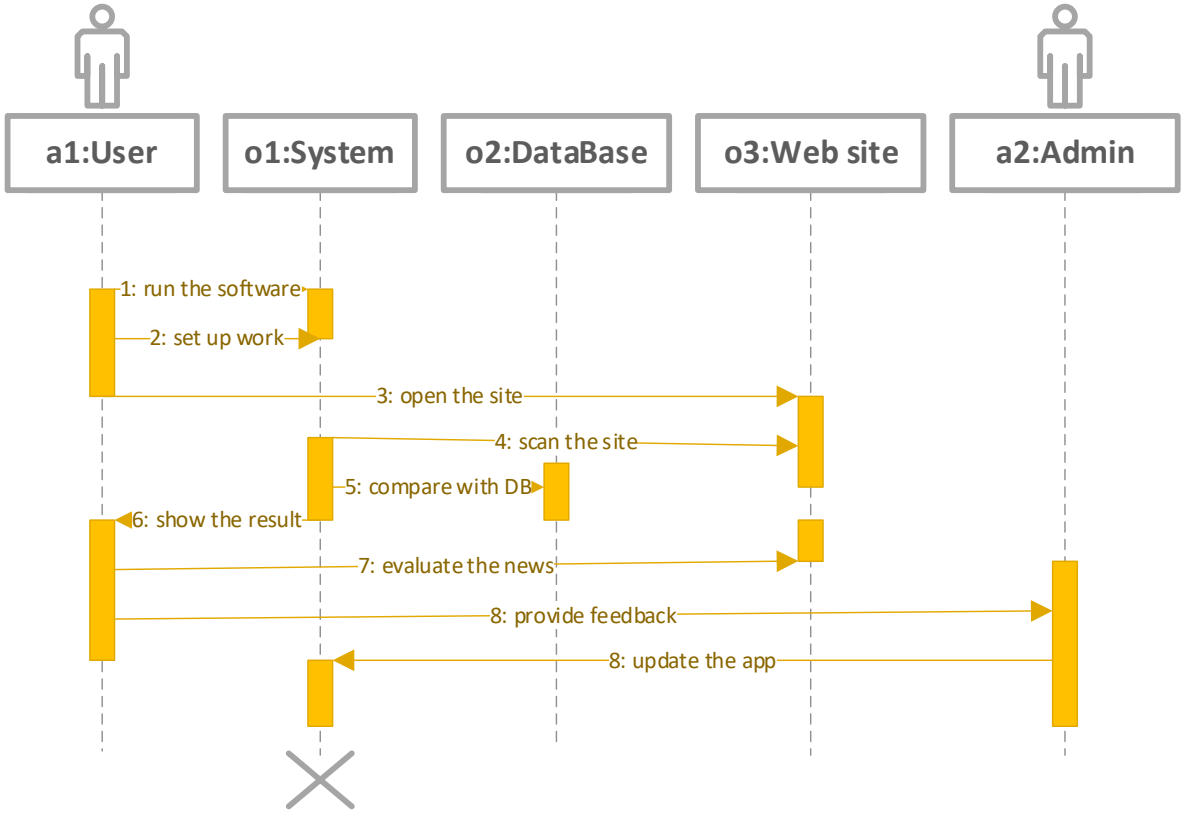

Figure 10: The flow chart for fake news checking system.

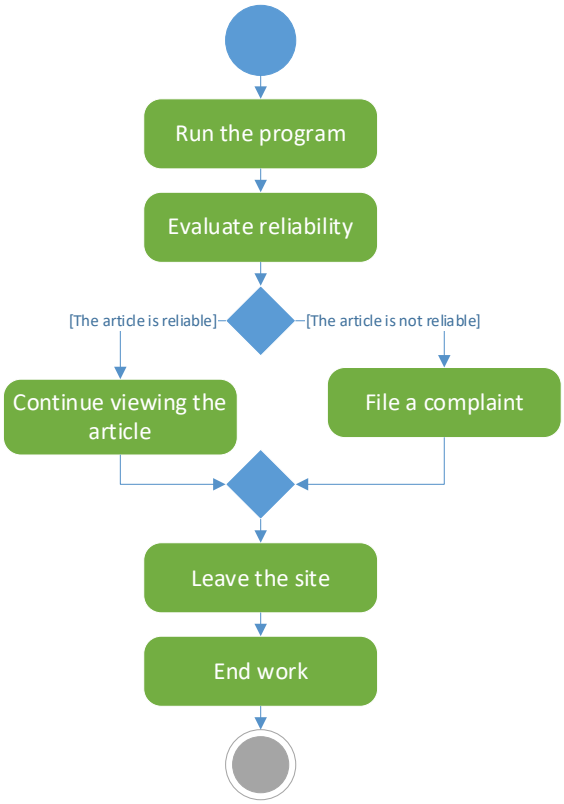

Figure 11: Activity diagram

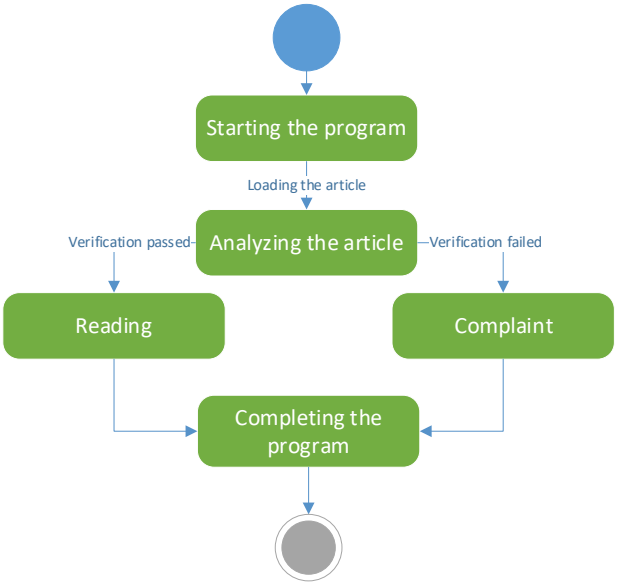

Figure 12: The state diagram

- Add the transition of setting the opening of the article from the state of Launching the program to the state of Analysis of the article.

- Add the check passed transition from the Article Analysis state to the Reading state.
- Add the transition of check not passed from the status of Analysis of the article to the state of Complaint.
- Add the transition to close the article to the Termination of work state.
- Add the transition from the Termination state of the program to the final state.

The deployment diagram has been developed for the spam detection system (Fig. 13).

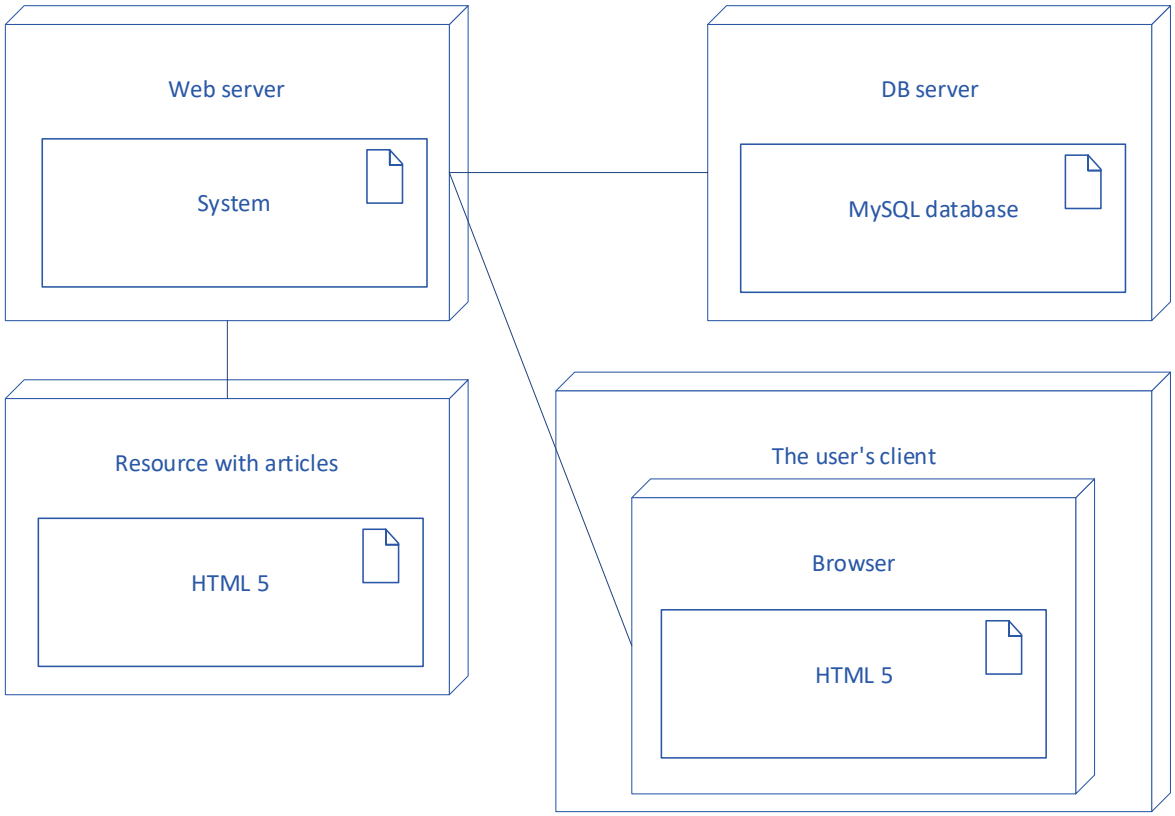

Figure 13: The deployment diagram.

Add nodes: System, DB, Resource with article and network with net stereotype.
It is possible to connect all nodes to the Global Network node.

5. Results and discussions

After developing the program, the folder with all the necessary files for work has been received. Among which is the main file with the program structure and algorithms. The separate file for localization and the separate file for the graphical interface of the program (Fig. 14-17).

Имя	Дата изменения	Тип	Размер
__pycache__	06.06.2022 14:36	Папка с файлами	
datasets	02.06.2022 14:41	Папка с файлами	
.gitignore	02.06.2022 14:40	Текстовый докум...	1 КБ
fake.py	03.06.2022 01:04	Файл "PY"	35 КБ
fake.ui	31.05.2022 23:46	Файл "UI"	14 КБ
fake_ui.py	31.05.2022 23:47	Файл "PY"	16 КБ
localization.py	02.06.2022 22:40	Файл "PY"	44 КБ
test.py	14.05.2021 23:34	Файл "PY"	11 КБ

Figure 14: The program folder.

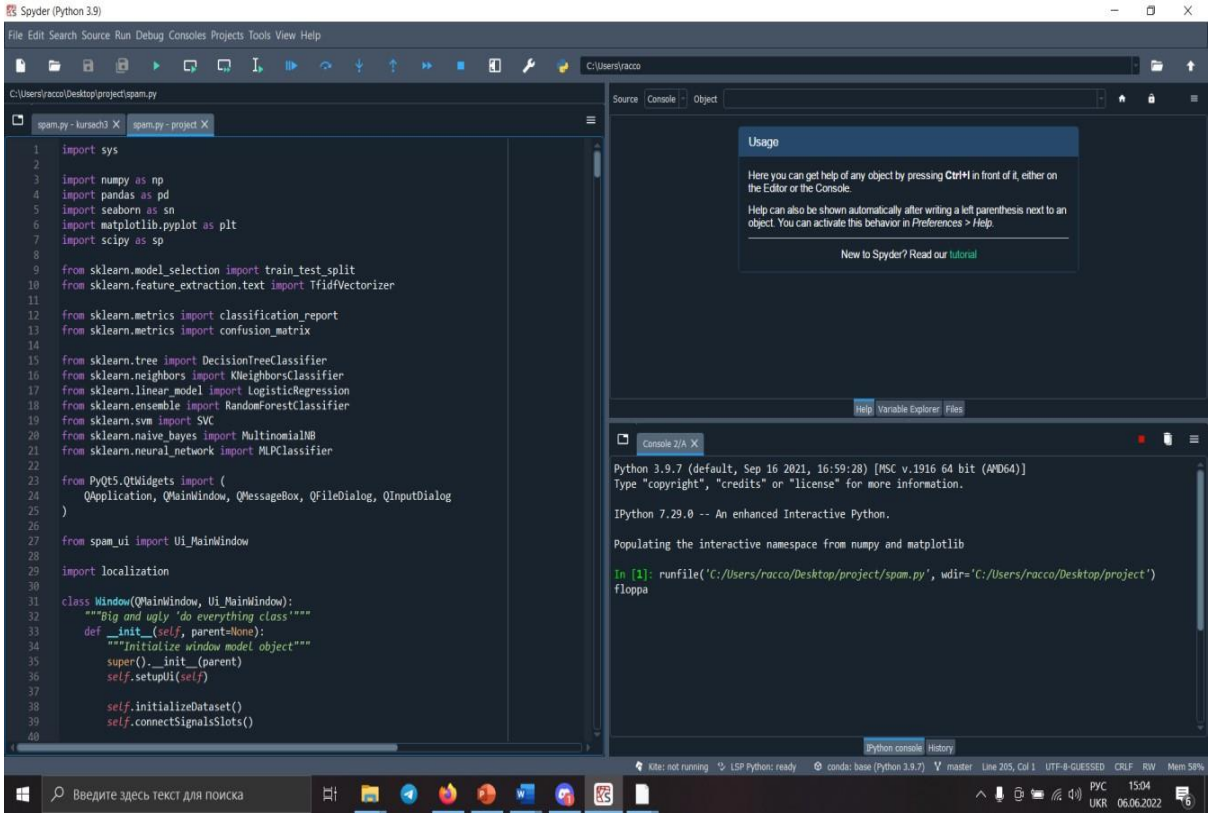

Figure 15: The environment in which the program was written

```
import sys

import numpy as np
import pandas as pd
import seaborn as sn
import matplotlib.pyplot as plt
import scipy as sp

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.neural_network import MLPClassifier

from PyQt5.QtWidgets import (
    QApplication, QMainWindow, QMessageBox, QFileDialog, QInputDialog
)

from spam_ui import Ui_MainWindow

import localization
```

Figure 16: Connecting everything necessary for the program

```
373     def drawDiagram(self):
374         """Draw diagrams"""
375         spam_text = self.localized('spam')
376         ham_text = self.localized('ham')
377 
378         confusion_matrix_text = self.localized('confusion-matrix')
379 
380         actual_text = self.localized('actual')
381         predicted_text = self.localized('predicted')
382 
383         # create dataframe with confusion matrix
384         df_cm = pd.DataFrame(
385             self.confusionMatrix,
386             index=[spam_text, ham_text],
387             columns=[spam_text, ham_text])
388 
389         # draw pretty heatmap using it
390         fig = plt.subplots()
391         plt.title(confusion_matrix_text)
392         ax = sn.heatmap(df_cm, annot=True, cmap = 'Wistia', fmt="d")
393         ax.set_xlabel(actual_text)
394         ax.set_ylabel(predicted_text)
395 
396 
397 
398         precision_text = self.localized('precision')
399         recall_text = self.localized('recall')
400         f1_score_text = self.localized('f1-score')
401         support_text = self.localized('support')
402 
403         labels = [precision_text, recall_text, f1_score_text, support_text]
404 
405         metrics = ['precision', 'recall', 'f1-score', 'support']
406 
407         fig, ax = plt.subplots(1, 4)
408 
409         # draw diagram for all of 4 model parameters comparing training and test scores
410         for i in range(4):
411             xs = [ham_text, spam_text]
412             ys = [self.report['ham'][metrics[i]], self.report['spam'][metrics[i]]]
```

Figure 17: The example of a function that draws diagrams

The program itself consists of two main screens, which are shown in Figure 18 and Figure 19. It is also possible to choose the language of the program from the general functions in the upper part of the screen (at the moment, Ukrainian and English are available for selection).

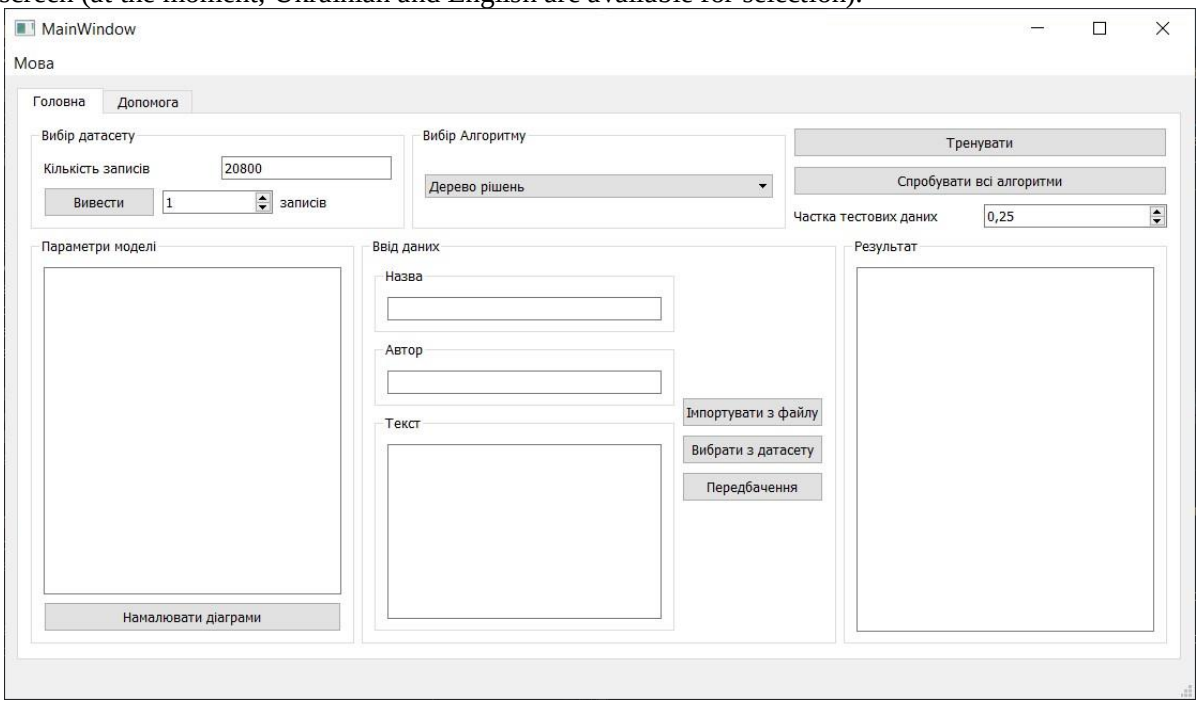

Figure 18: Main screen

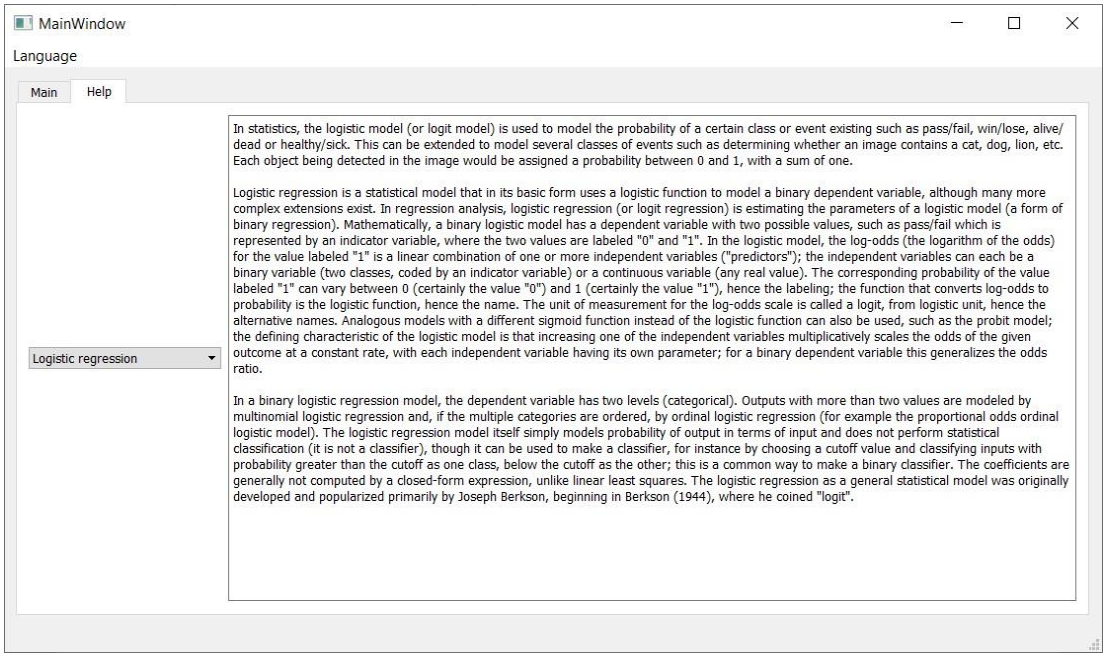

Figure 19: Help screen

The main functions are implemented on the first screen. Here it is possible to output a certain number of records from the dataset and choose exactly how many records will be used in the model. After that, it is needed to choose one of the seven algorithms that were described above. After that, it is possible to choose a partition for the training and test set (by default it is 75% to 25%) and train the model. Then, the results should appear in the window: accuracy for two sets, and parameters for the class fake and not fake. Also, after that, there is a possibility to build a confusion matrix and a graph with the parameters found. In addition, after training the model, it is possible to check how it works on the needed example. At this stage of development, there is a need to write a message to manually import from a file or select a specific record from a specific dataset. In the future, when the program is connected to the mail client, it will take data for verification from there.

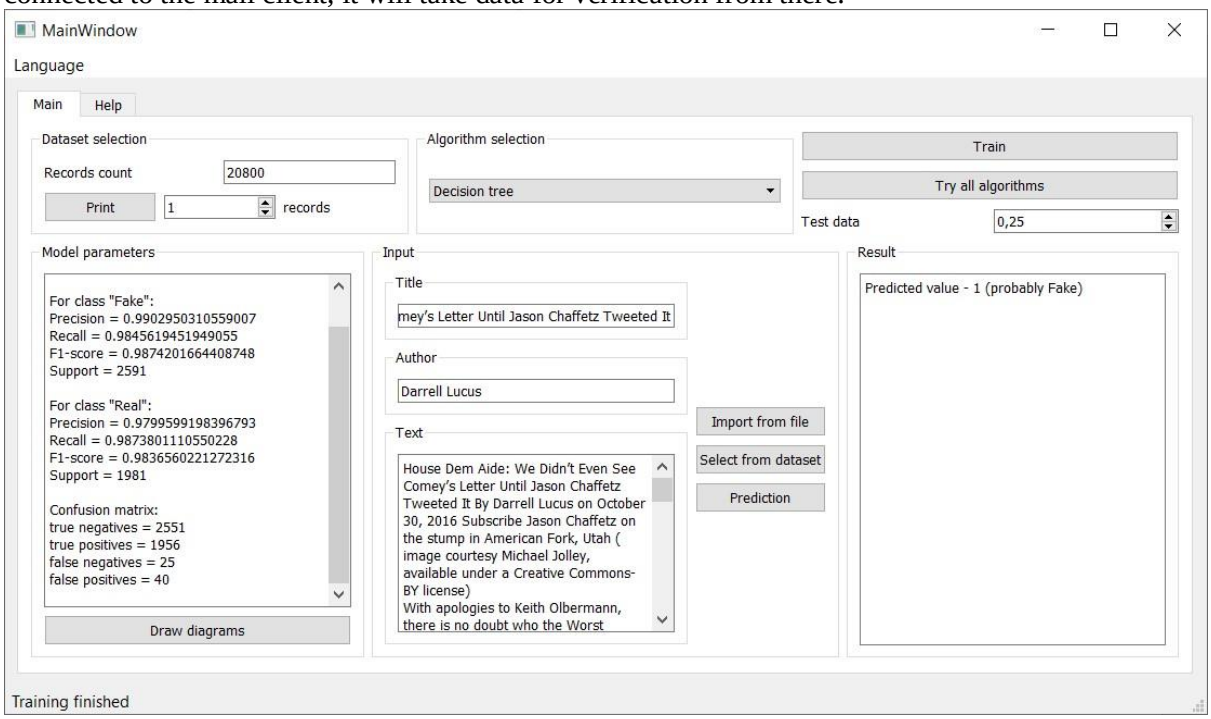

Figure 20: Functionality of the program

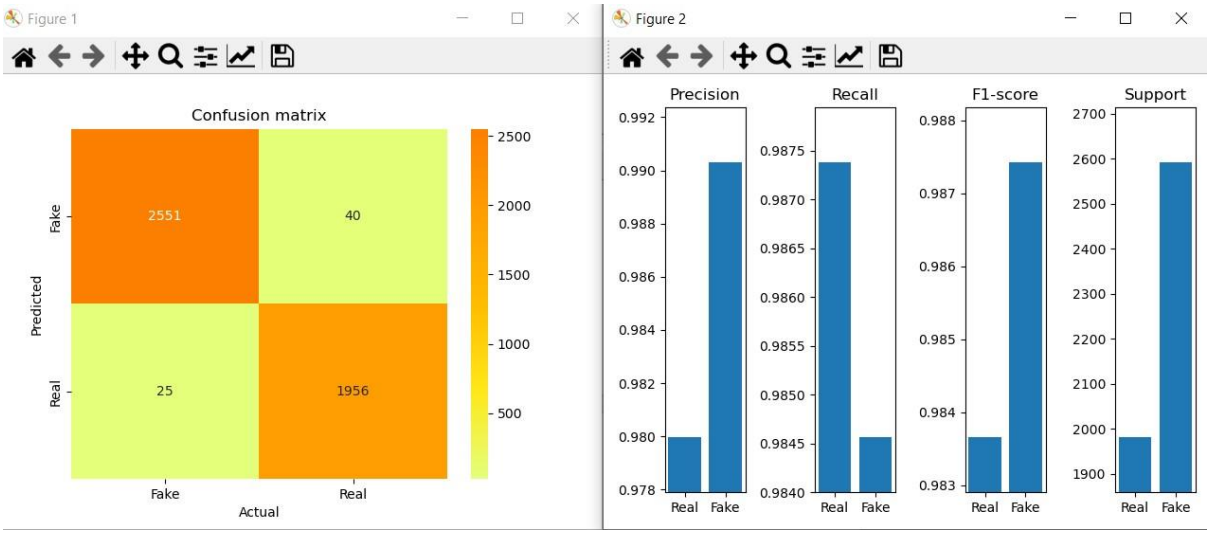

Figure 21: The matrix and graph built by the program using decision tree

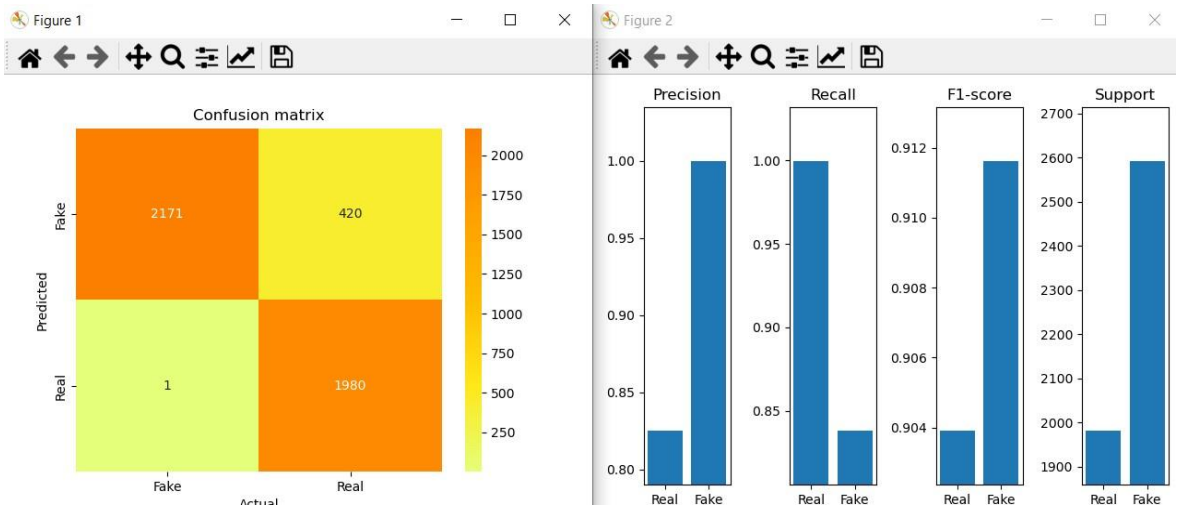

Figure 22: The matrix and graph built by program using the KNN

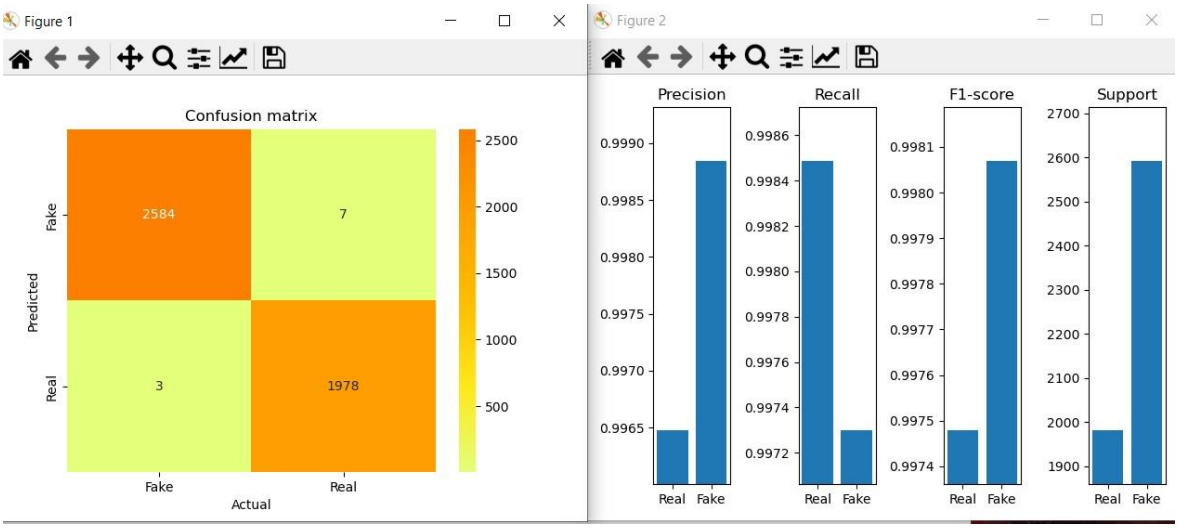

Figure 23: The matrix and graph built by the program using logistic regression program

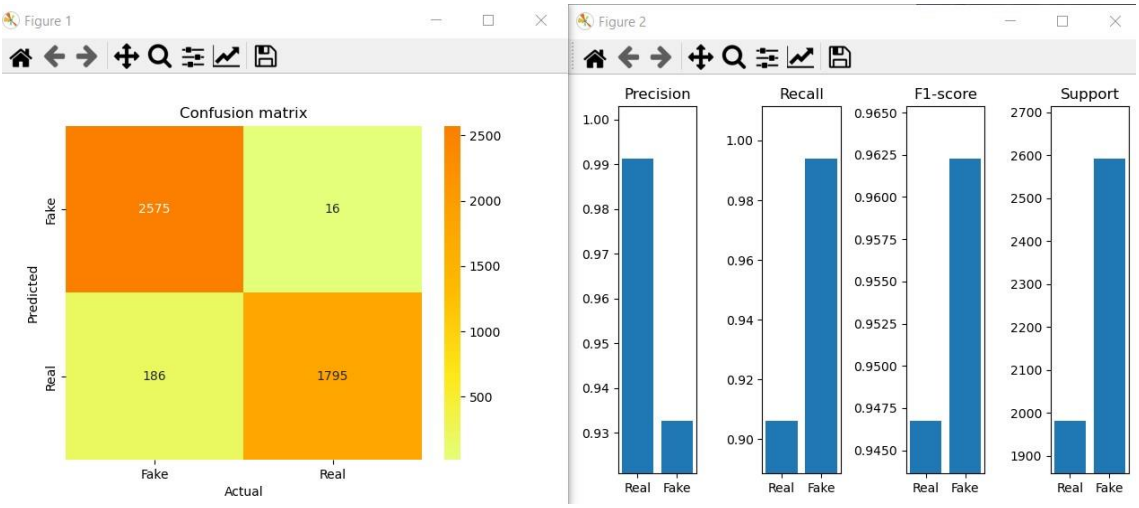

Figure 24: The matrix and graph are built by program using the random forest program

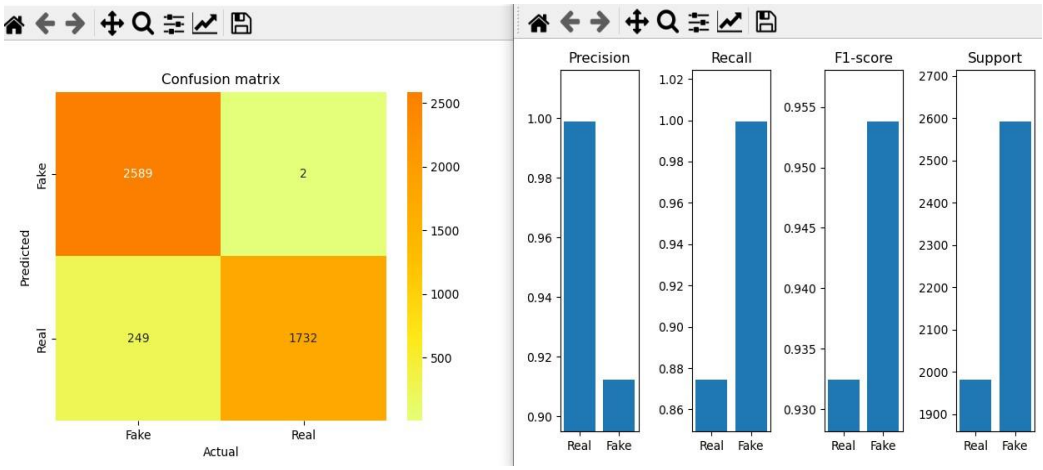

Figure 25: The matrix and graph are built by the program using Bayes

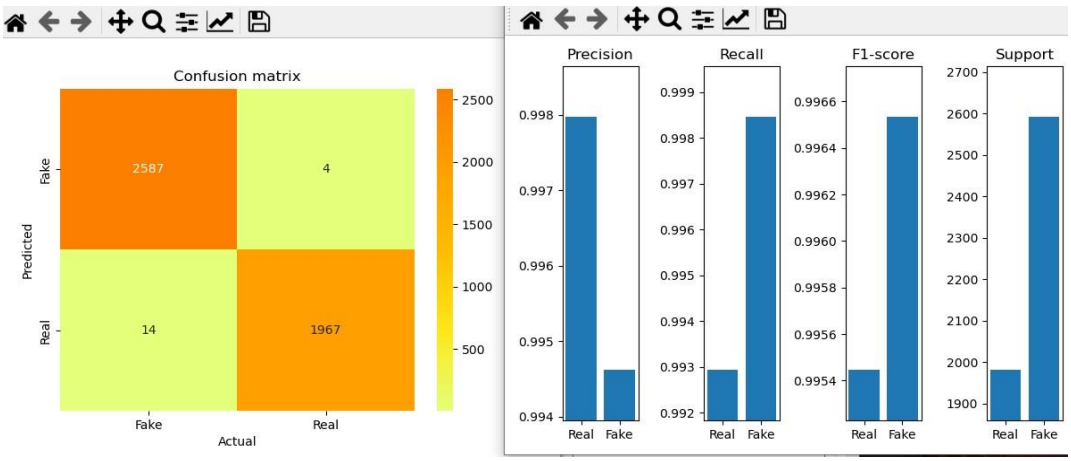

Figure 26: The matrix and graph are built by the program using a multilayer parceptron

So, six machine learning algorithms with data has been analyzed that give >90% accuracy. Some of them are better at recognizing fakes and some at recognizing real news.

Therefore, to improve the results, different methods should be combined.

- Using the decision tree, the accuracy for identifying fakes is 0.993, and non-fakes is 0.98 (difference 0.013).
- Using the KNN, the accuracy for identifying fakes is 0.999 and 0.83, respectively (difference 0.169).

- Using the logistic regression, the accuracy for identifying fakes is 0.9988 and 0.9965, respectively (difference 0.0023).
- Using the random forest, the accuracy for identifying fakes is 0.933 and 0.991, respectively (difference 0.058).
- Using the Bayes classifier, the accuracy for identifying fakes is 0.913 and 0.998, respectively (difference 0.085).
- Using the multilayer parceptron, the accuracy for identifying fakes is 0.9945 and 0.9979, respectively (difference 0.0034).

6. Conclusions

In the course of the work, the system to detect fake news via the Internet has been designed and created. For this, literary sources and scientific works related to this topic have been analyzed. Their analysis has confirmed that the issue of recognizing fake news via the Internet is very relevant and urgent in our time of global digitalization and Internet accessibility.

Similar solutions that are already on the market, have been analyzed: Bot Sentinel, CaptainFact, ClaimBuster, FakerFact, Hoaxy, NewsCheck, Our.news. Their strengths and weaknesses have been determined. Also, after the analysis, the functionality of the future system has been thought out and written. Therefore, the main advantages of this system over the considered analogues, its purpose, scope of application and possibilities for further development have been determined.

The business analysis has been carried out, the requirements and risks of the future startup have been determined. They also have drawn up a calendar and determined the necessary resources. Positive changes that project can bring to society have been identified.

The architectural and software solution has been chosen, which described the development of technologies and programming environments. Algorithms and datasets with which experiments have been conducted and selected in order visually to verify which of them are more suitable for which task.

As a result, the program has been developed, that with the most methods and datasets, gives results with an accuracy of >95%. It means that it copes well with its task and it is possible to make the final settings and test it in real conditions.

7. References

[1] S. I. Manzoor, J. Singla, Fake news detection using machine learning approaches: A systematic review. In 3rd international conference on trends in electronics and informatics (ICOEI), 2019, pp. 230-234.

[2] A. Jain, A. Shakya, H. Khatter, A. K. Gupta, A smart system for fake news detection using machine learning. In International conference on issues and challenges in intelligent computing techniques (ICICT), Vol. 1, 2019, September, pp. 1-4.

[3] Z. Khanam, B. N. Alwasel, H. Sirafi, M. Rashid, Fake news detection using machine learning approaches. In IOP conference series: materials science and engineering 1099 (2021) 012040.

[4] U. Sharma, S. Saran, S. M. Patil, Fake news detection using machine learning algorithms. International Journal of Creative Research Thoughts (IJCRT) 8(6) (2020) 509-518.

[5] P. Bahad, P. Saxena, R. Kamal, Fake news detection using bi-directional LSTM-recurrent neural network. Procedia Computer Science 165 (2019) 74-82.

[6] S. Ray, A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon), 2019, pp. 35-39.

[7] T. O. Ayodele, Types of machine learning algorithms. New advances in machine learning 3 (2010) 19-48.

[8] B. Mahesh, Machine learning algorithms-a review. International Journal of Science and Research (IJSR) 9 (2020) 381-386.

[9] F. A. Ozbay, B. Alatas, Fake news detection within online social media using supervised artificial intelligence algorithms. Physica A: statistical mechanics and its applications 540 (2020) 123174.

[10] Á. Ibrain Rodríguez, L. Lloret Iglesias, Fake news detection using Deep Learning. arXiv e-prints, arXiv-1910, 2019.
[11] S. Albota Semantic analysis of the Reddit vaccination news feed in the Reddit social network. In Computer science and information technologies: proceedings of IEEE 16th International conference CSIT, Lviv, Ukraine, 22–25 September, 2021, pp. 56–59.
[12] S. Albota War Implications in the Reddit News Feed: Semantic Analysis. In Computer science and information technologies: Int. conf. CSIT, Lviv, Ukraine, 10–12 November, 2022, pp. 99–102.
[13] I. Khomytska, V. Teslyuk, K. Prysyazhnyk, N. Hrytsiv, The Lehmann-Rosenblatt test applied for determination of statistical parameters of Charles Dickens's authorial style. In Computer Science and Information Technologies (CSIT): Proceedings of IEEE XVIth Scientific and Technical Conference, Lviv, Ukraine, 22–25 Sept. 2021, Vol. 2, pp. 64–67.
[14] I. Khomytska, V. Teslyuk, I. Bazylevych, Yu. Kordiiaka, Machine learning and classical methods combined for text differentiation, CEUR Workshop Proceedings 3171 (2022) 1107-1116.
[15] I. Khomytska, V. Teslyuk, I. Bazylevych, The statistical parameters of Ivan Franko's authorial style determined by the chi-square test. In Computer Science and Information Technologies (CSIT): Scientific and Technical Conference, Lviv, Ukraine, 10–12 November 2022, pp. 73–76.
[16] A. Roy, K. Basak, A. Ekbal, P. Bhattacharyya, A deep ensemble framework for fake news detection and multi-class classification of short political statements. In Proceedings of the 16th International Conference on Natural Language Processing, 2019, pp. 9-17.
[17] Y. Yang, L. Zheng, J. Zhang, Q. Cui, Z. Li, P. S. Yu, TI-CNN: Convolutional neural networks for fake news detection. arXiv preprint arXiv:1806.00749, 2018.
[18] L. Bozarth, C. Budak, Toward a better performance evaluation framework for fake news classification. In International AAAI conference on web and social media 14 (2020) 60-71.
[19] M. Choudhary, S. S. Chouhan, E. S. Pilli, S. K. Vipparthi, BerConvoNet: A deep learning framework for fake news classification. Applied Soft Computing 110 (2021) 107614.
[20] S. Ghosh, C. Shah, Towards automatic fake news classification. Proceedings of the Association for Information Science and Technology 55(1) (2018) 805-807.
[21] C. L. M. Jeronimo, L. B. Marinho, C. E. Campelo, A. Veloso, A. S. da Costa Melo, Fake news classification based on subjective language. In Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services, 2019, pp. 15-24.
[22] J. Kapusta, J. Obonya, Improvement of misleading and fake news classification for flective languages by morphological group analysis. Informatic 7(1) (2020) 4.
[23] J. V. de Souza, J. Gomes Jr, F. M. D. Souza Filho, A. M. D. Oliveira Julio, J. F. de Souza, A systematic mapping on automatic classification of fake news in social media. Social Network Analysis and Mining 10 (2020) 1-21.
[24] Y. Liu, Y. F. Wu, Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. Artificial intelligence 32(1), 2018.
[25] D. Mehta, A. Dwivedi, A. Patra, M. Anand Kumar, A transformer-based architecture for fake news classification. Social network analysis and mining 11 (2021) 1-12.
[26] R. K., Kaliyar, A., Goswami, P., Narang, S. Sinha, FNDNet–a deep convolutional neural network for fake news detection. Cognitive Systems Research 61 (2020) 32-44.
[27] Q. Li, Q. Hu, Y. Lu, Y. Yang, J. Cheng, Multi-level word features based on CNN for fake news detection in cultural communication. Personal and Ubiquitous Computing 24 (2020) 259-272.
[28] X. Zhi, L. Xue, W. Zhi, Z. Li, B. Zhao, Y. Wang, Z. Shen, Financial fake news detection with multi fact CNN-LSTM model. In Int. Conf. on Electronics Technology, 2021, pp. 1338-1341.
[29] B. Koloski, T. S. Perdih, M. Robnik-Šikonja, S. Pollak, B. Škrlj, Knowledge graph informed fake news classification via heterogeneous representation ensembles. Neurocomputing 496 (2022) 208-226.
[30] I. Kareem, S. M. Awan, Pakistani media fake news classification using machine learning classifiers. In 2019 International Conference on Innovative Computing (ICIC), 2019, pp. 1-6.
[31] M. Maree, M. Eleyat, Semantic Graph Based Term Expansion for Sentence-Level Sentiment Analysis. International Journal of Computing 19(4) (2020) 647-655. DOI:10.47839/ijc.19.4.2000
[32] N. A. Qarabash, H. A. Qarabash, Twitter Location-Based Data: Evaluating the Methods of Data Collection Provided By Twitter API. International Journal of Computing 19(4) (2020) 583-589. https://doi.org/10.47839/ijc.19.4.1992

[33] H. Lipyanina, A. Sachenko, T. Lendyuk, S. Nadvynychny, S. Grodskyi, Decision Tree Based Targeting Model of Customer Interaction with Business Page. CEUR Workshop Proceedings 2608 (2020) 1001-1012.
[34] 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.
[35] 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 Proceedings 2488 (2019) 369–382.
[36] V. Vysotska, S. Mazepa, L. Chyrun, O. Brodyak, I. Shakleina and V. Schuchmann, NLP Tool for Extracting Relevant Information from Criminal Reports or Fakes/Propaganda Content, In Int. Conf/ on Computer Sciences and Information Technologies (CSIT), 2022, pp. 93-98.
[37] M. Zanchak, V. Vysotska, S. Albota, The Sarcasm Detection in News Headlines Based on Machine Learning Technology. In Computer Sciences and Information Technologies (CSIT): proceedings of the IEEE 16th International Conference, 22-25 Sept., Lviv, Ukraine. 2021, pp. 131–137.
[38] V. Andrunyk, A. Vasevych, L. Chyrun, N. Chernovol, N. Antonyuk, A. Gozhyj, V. Gozhyj, I. Kalinina, M. Korobchynskyi, Development of information system for aggregation and ranking of news taking into account the user needs, CEUR Workshop Proceedings 2604 (2020) 1127–1171.
[39] S. Kubinska, R. Holoshchuk, S. Holoshchuk, L. Chyrun, Ukrainian Language Chatbot for Sentiment Analysis and User Interests Recognition based on Data Mining, CEUR Workshop Proceedings 3171 (2022) 315-327.
[40] A. Dmytriv, S. Holoshchuk, L. Chyrun, R. Holoshchuk, Comparative Analysis of Using Different Parts of Speech in the Ukrainian Texts Based on Stylistic Approach, CEUR Workshop Proceedings 3171 (2022) 546-560.
[41] N. Bondarchuk, I. Bekhta, Quantitative Characteristics of Lexical-Semantic Groups Representing Weather in Weather News Stories (Based on British Online Press), CEUR Workshop Proceedings 2870 (2021) 799-810.
[42] S. Orekhov, H. Malyhon, I. Liutenko, T. Goncharenko, Using Internet News Flows as Marketing Data Component, CEUR workshop proceedings 2604 (2020) 358-373.
[43] S. Albota, Linguistic and Psychological Features of the Reddit News Post, In Int. Scientific and Technical Conf. on Computer Sciences and Information Technologies, CSIT, 2020, pp. 295–299.
[44] N. Shakhovska, M. Medykovskyj, 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.
[45] N. Liubchenko, A. Podorozhniak, V. Oliinyk, Research Application of the Spam Filtering and Spammer Detection Algorithms on Social Media, CEUR Workshop Proceedings 3171 (2022) 116-126.
[46] K. Datsyshyn, Z. Haladzhun, N. Kunanets, O. Hotsur, N. Veretennikova, Neologisms with the Prefix Anti- in the Ukrainian Online Media in the Covid-19 Pandemic Period, CEUR Workshop Proceedings 3171 (2022) 192-211.
[47] O. Romanovskyi, N. Pidbutska, A. Knysh, E. Vorobieva, Leadership Detection Across Social Media Hashtags, CEUR Workshop Proceedings 3171 (2022) 632-641.
[48] O. Yurchenko, N. Ugolnikova, Linguistic Methods in Social Media Marketing, CEUR Workshop Proceedings 2870 (2021) 743-754.
[49] Z. Haladzhun, O. Harmatiy, Y. Bidzilya, N. Kunanets, K. Shunevych, Hate Speech in Media Towards the Representatives of Roma Ethnic Community, CEUR Workshop Proceedings 2870 (2021) 755-768.
[50] N. Romanyshyn, Application of Corpus Technologies in Conceptual Studies (based on the Concept Ukraine Actualization in English and Ukrainian Political Media Discourse), CEUR workshop proceedings 2604 (2020) 472-488.
[51] 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.