Специфікація вимог до програмного забезпечення для програмної системи

«Fake News Detection Models»  
Затверджено версією 1.0  
Підготовлено Шагуном Артемом Сергійовичем  
Харківський національний університет радіоелектроніки  
17.06.2025

1. Вступ

1.1. Мета

Цей документ визначає функціональні та нефункціональні вимоги до додатку «Fake News Detection Models».

1.2. Умовні позначення

Ключові компоненти виділяються жирним, приклади та змінні – курсивом, назви функцій, моделей і змінних – у стилі snake\_case або CamelCase.

1.3. Цільова аудиторія

дослідники в галузі NLP, студенти, розробники Python, спеціалісти з машинного навчання.

1.4. Область проєкту

Програмну систему призначено для виявлення фейкових новин за допомогою моделей машинного навчання. Передбачено попередню обробку тексту, векторизацію та застосування моделей Logistic Regression, SVM, XGBoost, BERT для класифікації новин як правдивих або фейкових.

1.5. Посилання

* IEEE Std 830-1998;
* ISO/IEC 25010:2011.

2. Загальний опис

2.1. Перспектива продукту

Програма є консольною Python-утилітою, яка реалізує процес виявлення фейкових новин з можливістю порівняльного аналізу моделей.

2.2. Основні функції

* завантаження збереженої ML-моделі та векторизатора;
* попередня обробка введеного тексту (очищення, токенізація);
* векторизація тексту;
* класифікація новини як фейкової або правдивої;
* отримання й вивід ймовірностей класифікації;
* обчислення метрик точності, recall, precision, F1;
* збереження результатів експериментів;
* порівняння моделей по точності.

2.3. Користувачі

Усі користувачі мають однаковий функціонал.

2.4. Оточення

* Python: >=3.10;
* Бібліотеки: scikit-learn, nltk, spacy, transformers, joblib;
* ОС: Linux, Windows, macOS.

2.5. Обмеження

* підтримка лише англомовних новин;
* використання лише заздалегідь натренованих моделей;
* CLI-інтерфейс без GUI.

2.6. Документація

* README у репозиторії.

2.7. Припущення

* користувач має базові знання Python;
* встановлено необхідні бібліотеки.

3. Зовнішні інтерфейси

3.1. Інтерфейс користувача

* CLI-інтерфейс: введення тексту або шляху до файлу;
* вивід результатів у консоль.

3.2. Апаратні інтерфейси

* комп’ютер користувача.

3.3. Програмні інтерфейси

* моделі збережені у форматі joblib;
* взаємодія з бібліотеками ML/NLP.

3.4. Комунікаційні інтерфейси

* відсутні, програмна система працює локально.

4. Особливості системи

4.1. Моделі класифікації

* Logistic Regression
* Support Vector Machine
* XGBoost

4.2. Проведення тестів

* запуск тестів з реального датасету;
* виведення метрик (accuracy, precision, recall, F1-score);
* логування результатів.

4.3. Вивід результатів

* класифікація новини (Fake/Real);
* ймовірності по класах;
* результати порівняння моделей.

4.4. Налаштування

* вибір моделі;
* шлях до моделі та векторизатора.

5. Нефункціональні вимоги

5.1. Продуктивність

* класифікація тексту повинна виконуватись у межах 1–2 секунд для моделей LR/SVM/XGBoost.

5.2. Безпека

* відсутнє збереження персональних даних;
* відсутність інтернет-з'єднання.

5.3. Якість

* відповідність PEP8;

5.4. Надійність

* коректна обробка помилок підключення та виконання;
* вивід помилок у консоль.

5.5. Гнучкість

* можливість легко додавати нові моделі;
* заміна векторизатора без зміни структури програмної системи.

Appendix A: Глосарій

* NLP – Natural Language Processing
* ML – Machine Learning
* Vectorizer – засіб перетворення тексту в числовий вигляд

Appendix B: Моделі аналізу

* схема класифікації тексту (див. додатки до дипломної роботи

Appendix C: TBD

* підтримка багатомовності
* розширення на онлайн-детекцію через API