Боднарчук Сергій Ігорович, бакалавр

Чернівецький національний університет імені Ю. Федьковича, Чернівці

**АВТОМАТИЧНА КЛАСИФІКАЦІЯ ЗАДАЧ З ВИКОРИСТАННЯМ ЗАСОБІВ ML.NET**

У наш час створюється, розвивається і підтримується дуже багато проектів і стартапів. Здебільшого вони кардинально відрізняються підходами до розробки, технологіями, розмірами команд. Проте у всіх проектів є спільна риса – вони використовують системи управління проектами, обов’язково частиною яких є системи відстежування помилок (багтрекери), такі як Jira, Bugzilla, GitHub [1].

Оскільки з ростом проекту збільшується і кількість задач, які мають бути вирішені, на їх створення і класифікацію витрачається все більше і більше часу. Головними перевагами автоматичної класифікації є:

1. Підвищення ефективності роботи – розробники витрачають менше часу на опис задачі.
2. Точність класифікації – методи машинного навчання дозволяють класифікувати задачі з високою точністю.
3. Мінімізація людської помилки – робота з описом задач є дуже монотонною і через це часто приводить до помилок.

Для вирішення подібних задач чудово підходить машинне навчання, оскільки модель створюється на основі зразків [2]. У нашому випадку зразками можуть бути задачі створені до впровадження автоматичної класифікації.

Метою роботи є створення навченої моделі здатної класифікувати задачі у системах відстежування помилок. Щоб працювати з моделлю було створено .Net додаток. NET - це середовище розробки програмного забезпечення, створене корпорацією Майкрософт, яке дозволяє розробникам створювати широкий спектр додатків, включаючи веб-додатки, десктопні, мобільні, ігрові, додатки для Інтернету речей та штучного інтелекту. Вона надає середовище виконання, а також набір бібліотек та інструментів для створення та розгортання додатків на різних платформах [3]. Для створення моделей використовувалась бібліотека ML.NET. Центральним елементом ML.NET є модель машинного навчання. Модель визначає кроки, необхідні для перетворення вхідних даних у прогноз. За допомогою ML.NET можна навчити власну модель, вказавши алгоритм, або імпортувати попередньо навчені моделі TensorFlow і ONNX.

Для навчання моделі важливим етапом є вибір алгоритму (стратегії) навчання. Оскільки задачі можуть бути 2 і більше типів, тому потрібно використовувати багатокласову класифікацію (multi-class classification). Було вирішено використовувати алгоритм one-vs-rest з використанням алгоритму бінарної класифікації averaged perceptron для кожного окремого класифікатора.

В алгоритмі one-vs-rest для набору n екземплярів класів потрібно згенерувати n бінарних класифікаторів. Кожен бінарний класифікатор навчається окремо і буде передбачати ймовірність для свого екземпляру класу [4]. При передбачені буде отримано екземпляр того класу, бінарний класифікатор якого повернув найбільше значення.

Результатом роботи є навчена модель для класифікації задач та додаток, котрий дозволяє працювати з моделлю.

**Література**

1. Багтрекер: [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B3%D1%82%D1%80%D0%B5%D0%BA%D0%B5%D1%80
2. Machine learning: [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Machine\_learning
3. What is .NET? Introduction and overview: [Електронний ресурс] – Режим доступу: https://learn.microsoft.com/en-us/dotnet/core/introduction
4. Multi-class classification – One-vs-all and One-vs-one: [Електронний ресурс] – Режим доступу: <https://towardsdatascience.com/multi-class-classification-one-vs-all-one-vs-one-94daed32a87b>