Московский государственный технический университет имени Н. Э. Баумана
(национальный исследовательский университет)

Обнаружение дефектов программного обеспечения с использованием алгоритмов машинного обучения

Студент: Климов Илья Сергеевич, ИУ7-82Б
Научный руководитель: Вишневская Татьяна Ивановна

Москва, 2023

Цель и задачи

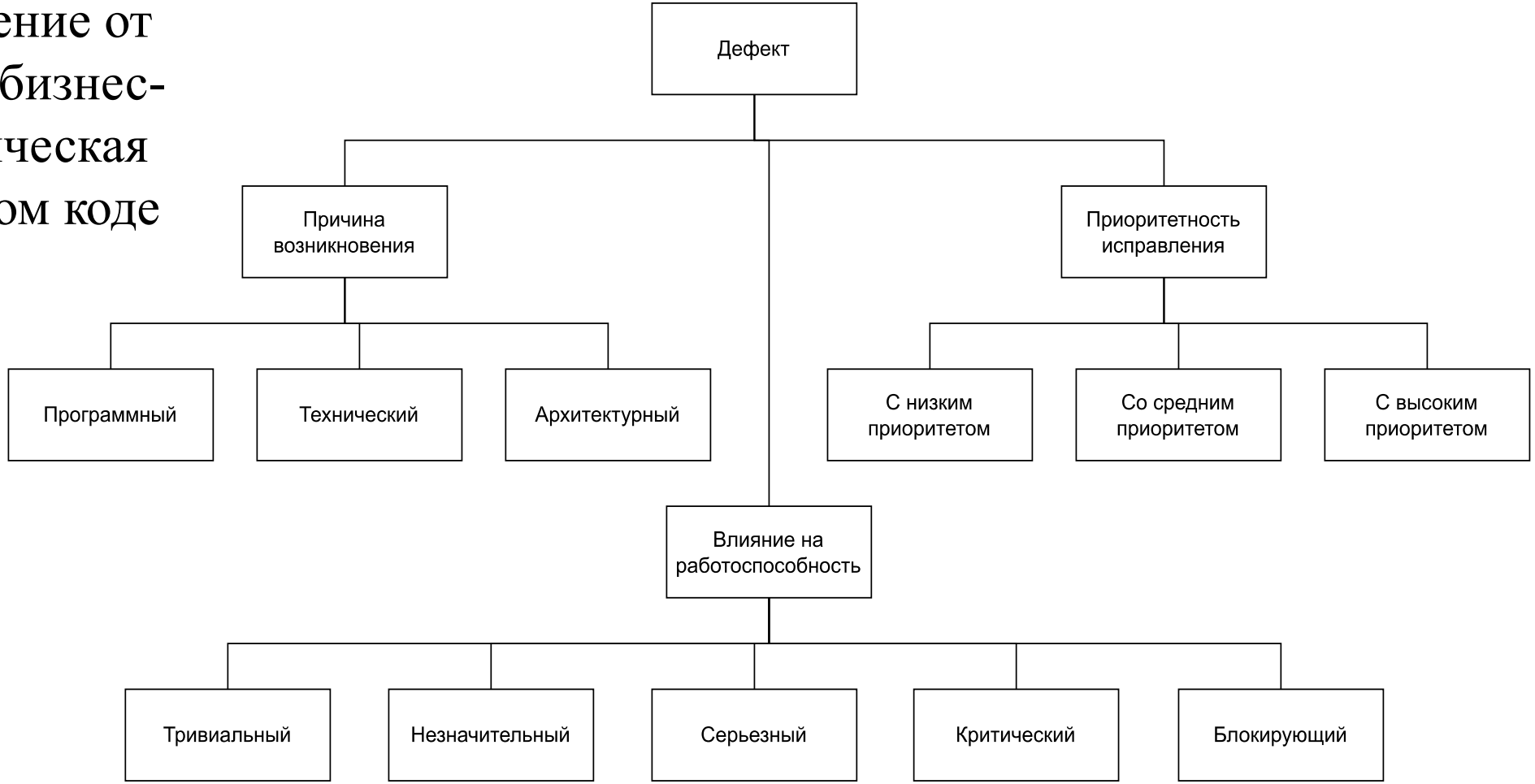
Цель: разработка и программная реализация метода обнаружения дефектов ПО с использованием алгоритмов машинного обучения.

Задачи:

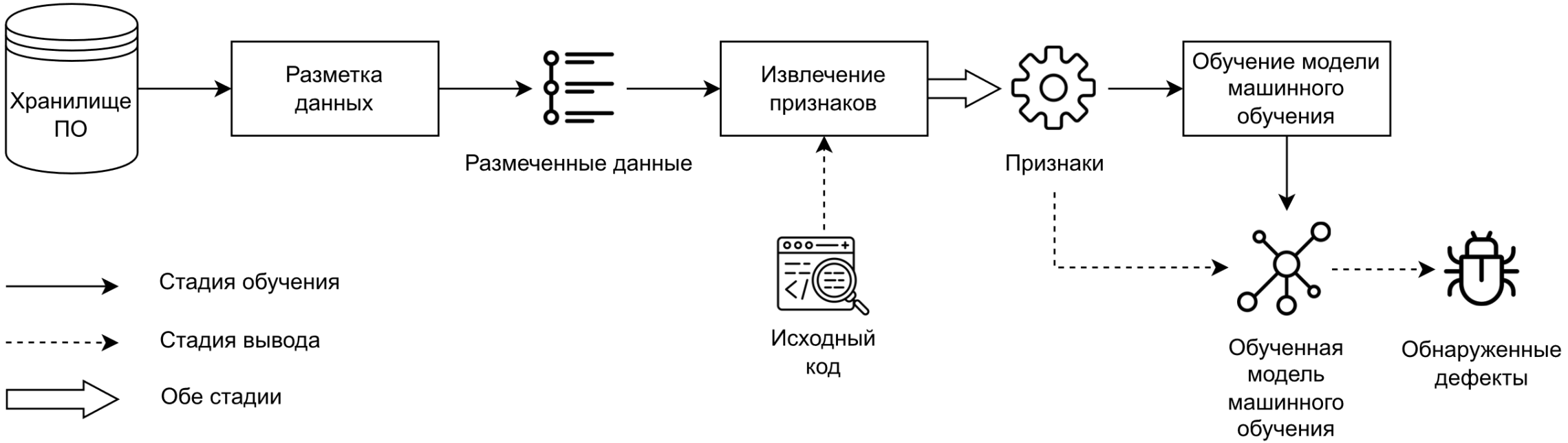
1) проанализировать и сравнить существующие методы машинного обучения для обнаружения дефектов ПО;
2) разработать метод обнаружения дефектов ПО с применением ансамбля деревьев решений (градиентного бустинга);
3) разработать программное обеспечение, реализующее метод обнаружения дефектов ПО;
4) провести исследование эффективности разработанного метода и сравнение его с существующими реализациями.

Дефекты разрабатываемого ПО

Дефект – отклонение от первоначальных бизнес-требований, логическая ошибка в исходном коде программы.



Процесс обучения модели обнаружения дефектов ПО



Метрики для сравнения алгоритмов

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

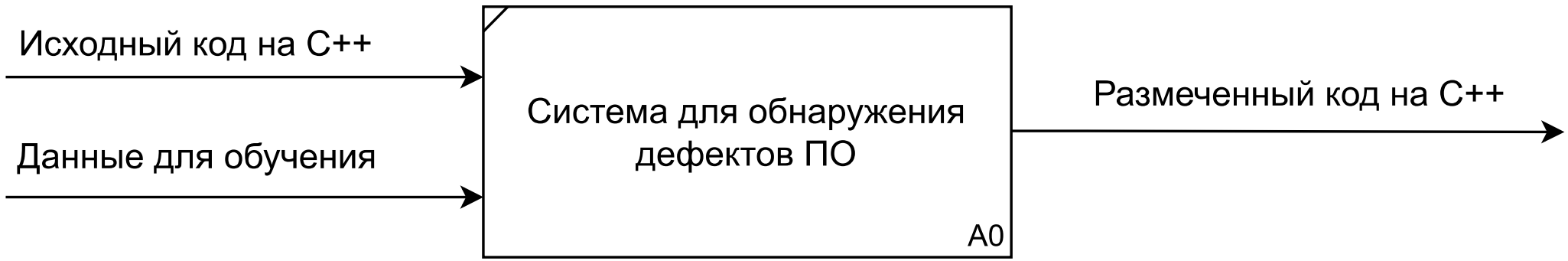
$$F - \text{measure} = \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

TP – объекты, которые были верно классифицированы как положительные;
TN – объекты, которые были верно классифицированы как отрицательные;
FP – объекты, которые были ложно классифицированы как положительные;
FN – объекты, которые были ложно классифицированы как отрицательные.

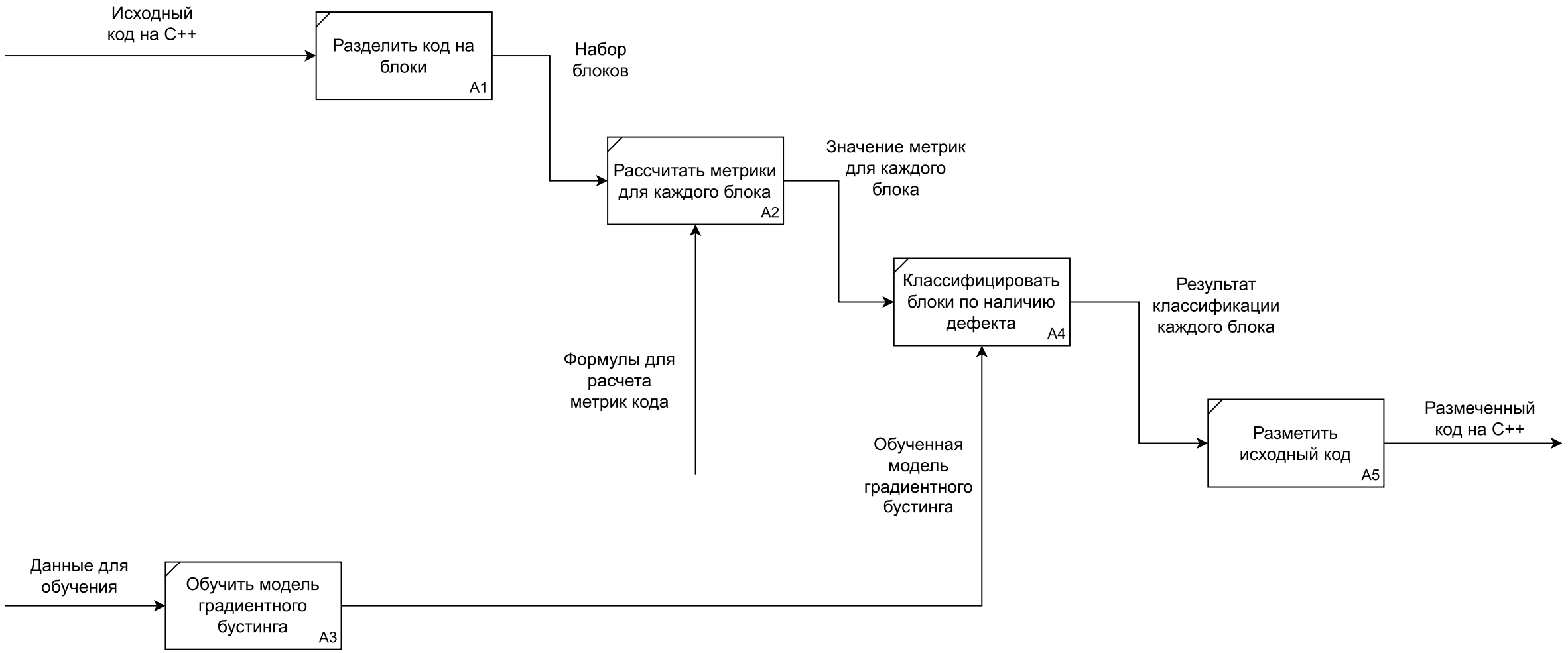
Сравнительная таблица результатов работы алгоритмов

Метрики / Алгоритмы	**Accuracy (точность)**	**Precision (точность)**	**Recall (полнота)**	**F-measure (F-мера)**
Наивный байесовский классификатор	0.795	0.845	0.803	0.849
Метод опорных векторов	0.841	0.901	0.879	0.902
Дерево решений	0.823	0.845	0.878	0.889
Алгоритм случайного леса	0.845	0.859	0.863	0.890
Градиентный бустинг	**0.847**	**0.903**	**0.883**	**0.903**
Адаптивный бустинг	0.835	0.858	0.861	0.889

Формализованная постановка задачи



Метод обнаружения дефектов ПО



Обучение модели градиентного бустинга

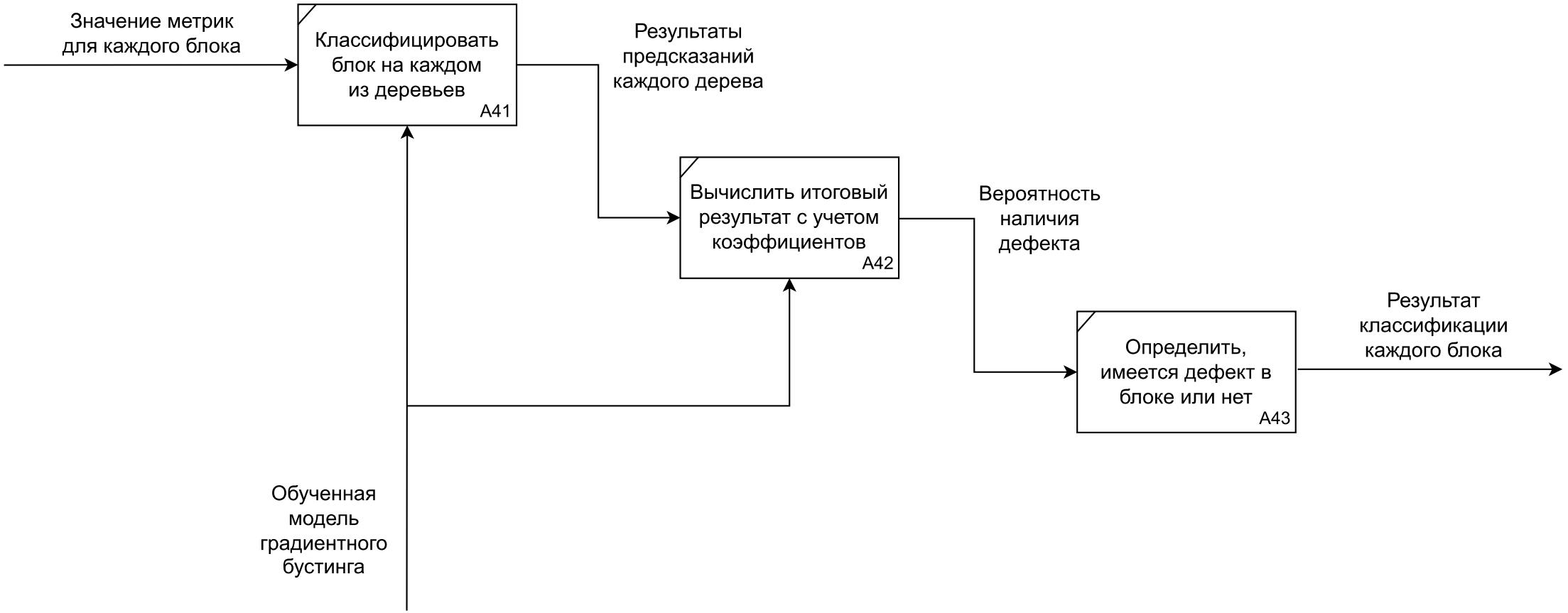
Критерий Джини: $H(R) = 2p_+(1 - p_+)$

Модель градиентного бустинга: $a_k(x) = \eta b_1(x, y_1) + \eta b_2(x, y_2) + \cdots + \eta b_k(x, y_k)$

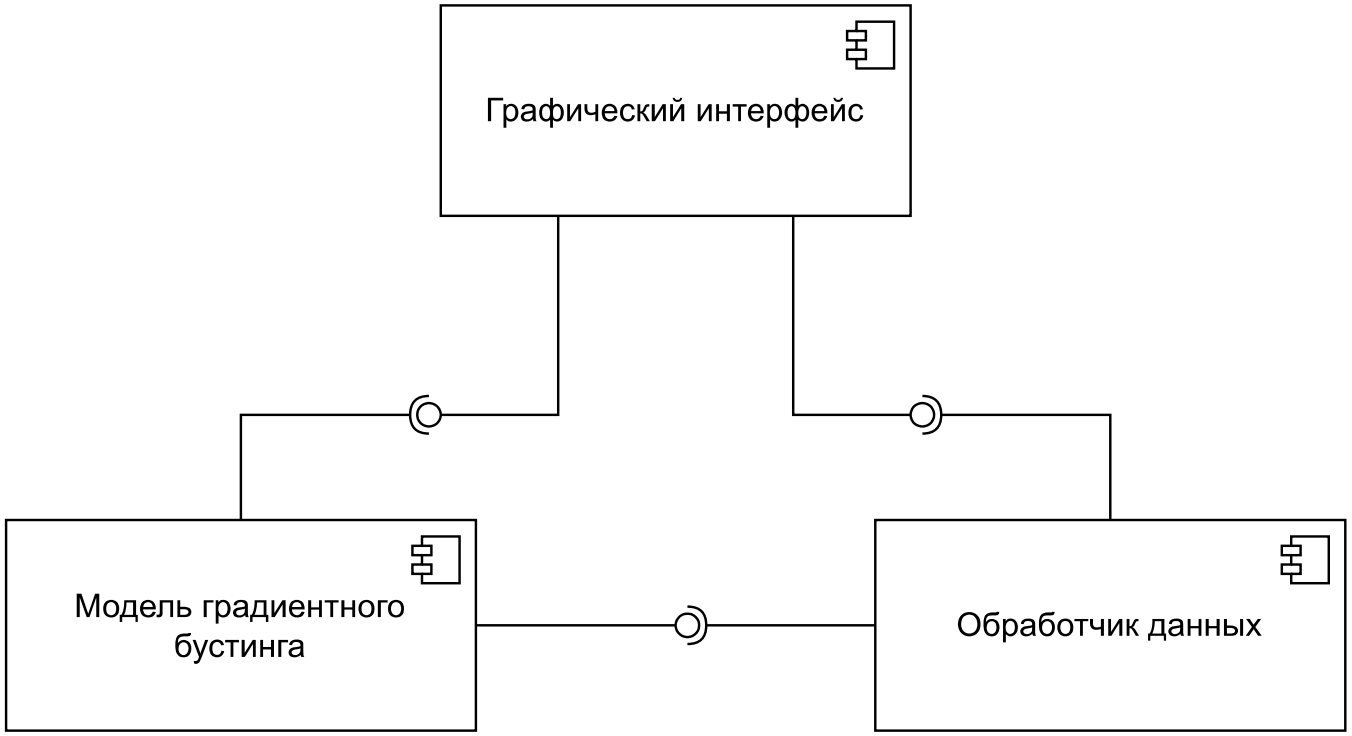
Логарифмическая функция потерь: $\mathcal{L}(y, p) = -(y \cdot \log(p) + (1 - y) \cdot \log(1 - p))$

Антиградиент: $g_i^k = -\frac{a_k(x_i) - y_i}{a_k(x_i)(a_k(x_i) - 1)}$

Классификация кода по наличию дефекта



Программная реализация разработанного метода



Язык программирования – Python.

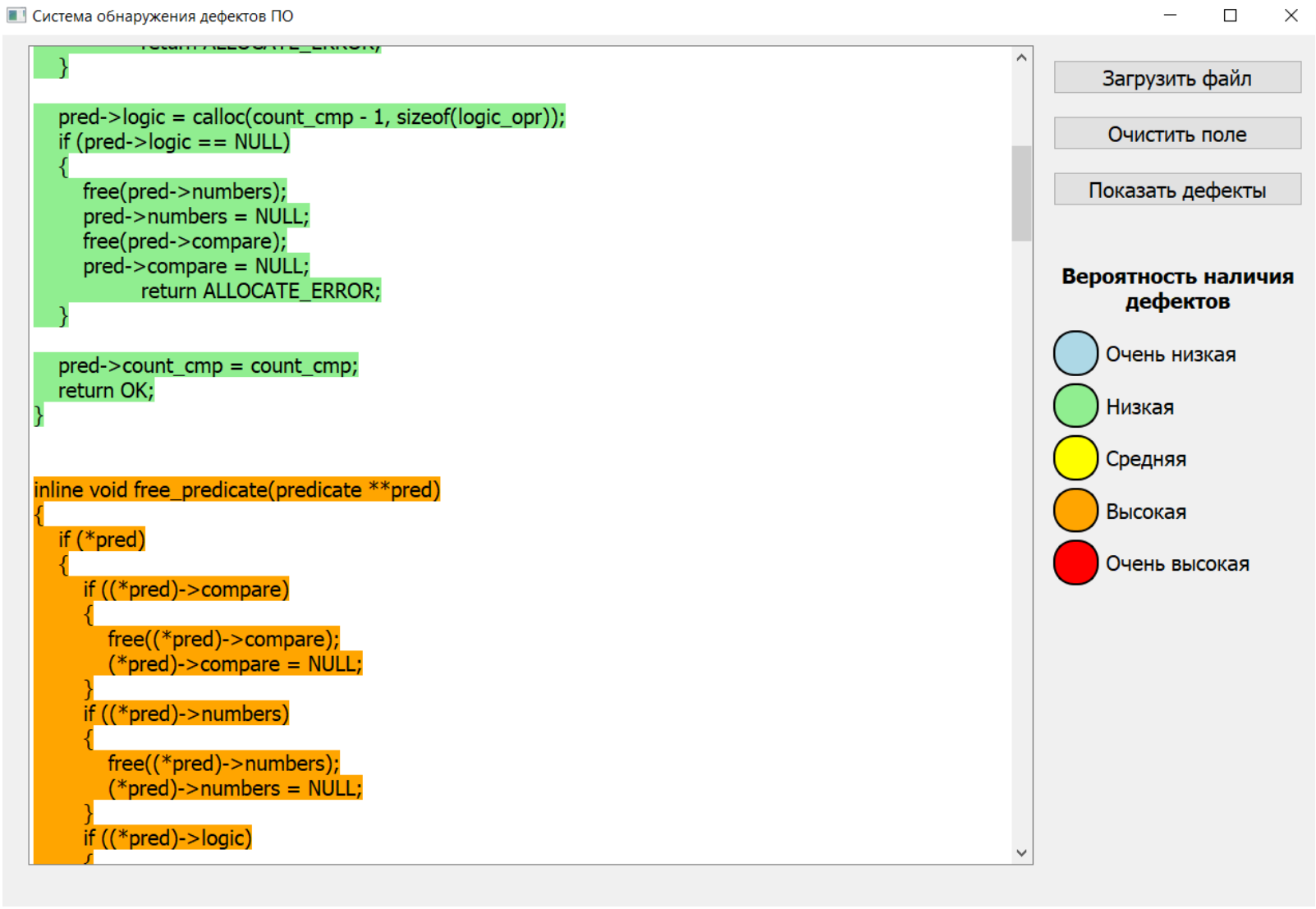
Библиотека для предобработки данных – scikit-learn.

Библиотека для обучения модели методом градиетного бустинга – XGBoost.

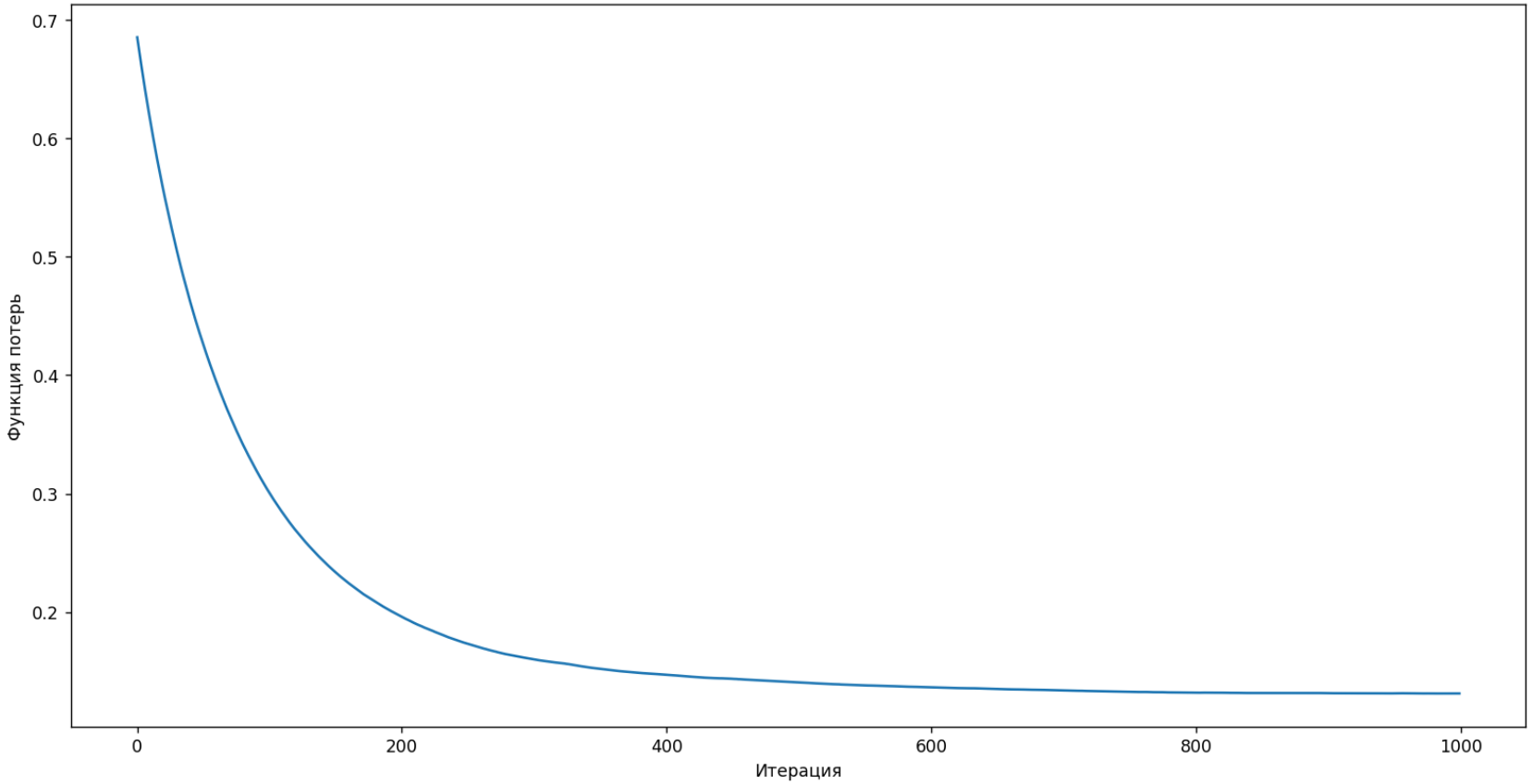
Фреймворк для разработки приложения – PyQt5.

Среда разработки – PyCharm.

Пример работы программы



Анализ точности метода



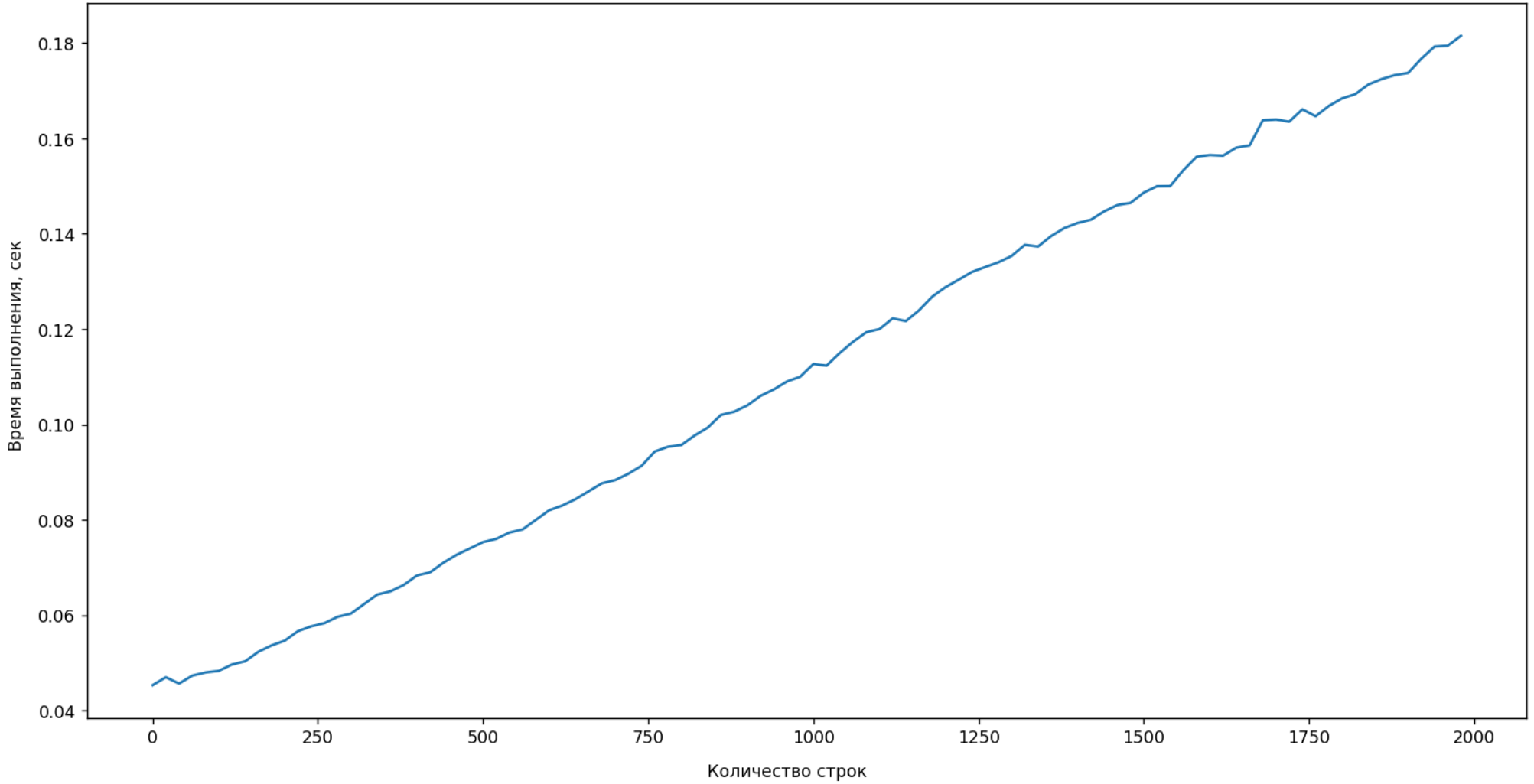
Accuracy = 0.941

Precision = 0.818

Recall = 0.936

F-measure = 0.873

Анализ времени выполнения программы



Заключение

В результате выполнения работы достигнута цель и решены все поставленные задачи:

1) проанализированы и сравнены существующие методы машинного обучения для обнаружения дефектов ПО;
2) разработан метод обнаружения дефектов ПО с применением ансамбля деревьев решений (градиентного бустинга);
3) разработано программное обеспечение, реализующее метод обнаружения дефектов ПО;
4) проведено исследование эффективности разработанного метода и сравнение его с существующими реализациями.