**«Кибернетика»**

Направление подготовки 01.03.0

**Пояснительная записка** к курсовой работе на тему:

Система оптимальной доразметки данных в задачах классификации изображений

Группа Б20-504, Б20-514Б15-501

Дмитрий

Студент Фонов Н.Д.

Сергей

Гудкова Д.А.

Павел

Каретникова О.А.

Семенова Ирина

Мальгина Э.А.

Але Фигурин И.И.

|  |  |  |  |
| --- | --- | --- | --- |
|  | (подпись) |  | (ФИО) |
| Преподаватель |  |  | Ди Пирова |

Сергеевич

(подпись) (ФИО)

Итоговая оценка (0-100 баллов) ECTS

**Москва 2018**

Содержание

[Введение 3](#_Toc148688839)

[Функциональные требования системы 5](#_Toc148688840)

[Выбор модели жизненного цикла 6](#_Toc148688841)

[Диаграмма прецедентов 8](#_Toc148688842)

[Диаграмма состояний 11](#_Toc148688843)

[Диаграмма классов 12](#_Toc148688844)

[Сценарии работы 13](#_Toc148688845)

[Диаграммы последовательностей 19](#_Toc148688846)

[Краткое руководство пользователя 19](#_Toc148688847)

[План тестирования 19](#_Toc148688848)

[Тест-кейсы 19](#_Toc148688849)

[План интеграции 19](#_Toc148688850)

[Сопровождение ПО 19](#_Toc148688851)

# Введение

Название проекта:

Система оптимальной доразметки данных в задачах классификации изображений.

Цель проекта:

Целью проекта является разработка ПО, обеспечивающего оптимизацию процесса разметки изображений с помощью искусственного интеллекта, используя методы активного обучения. При разработке данного ПО будут оптимизироваться такие показатели, как корректность и стоимость разметки данных, а также точность классификатора, обученного на размеченных данных.

Модель жизненного цикла:

Agile-модель.

Достоинства:

* Гибкость и открытость к любым изменениям.
* Сниженные риски провала.
* Устойчивость к срыву сроков.
* Большая вовлечённость команды.
* Высокая скорость реакции на проблемы.
* Минимум рутины.

Основные требования:

* Загрузка датасета с частичной разметкой или без нее
* Выбор алгоритма предобработки изображений
* Выбор метода кросс-валидации
* Выбор ML-алгоритма для обучения
* Доразметка данных
* Выгрузка обученной ML-модели
* Просмотр метрик качества классификации

Выбор инструментальных средств проектирования и разработки:

* **Electron** - это фреймворк для разработки настольных приложений с использованием HTML, CSS и JavaScript. Такие приложения могут работать на различных платформах. Среди них — Windows, Mac и Linux.
* **PyCharm** - это интегрированная среда разработки на языке Python, созданная компанией JetBrains.
* **Visual Paradigm Online** –онлайн-инструменты для совместной обработки диаграмм, сопоставления карт клиентов и управления задачами. С помощью решения можно быстро создавать профессиональные диаграммы и хранить проекты и диаграммы команды в облаке. Диаграммы Решение включает различные типы диаграмм (общие типы, блок-схемы, UML, бизнес-карты, AWS и т. д.) и более 500 примеров готовых проектов.

# Функциональные требования системы

1. Программа должна иметь функцию загрузки частично размеченного или неразмеченного датасета в виде архива\* изображений;
2. Программа должна иметь функцию выгрузки обученной ML-модели в формате .onnx;
3. Программа должна предоставлять возможность просматривать файлы в одной из папок архива датасета,
4. Программа должна предоставлять пользователю возможность выбора алгоритма предобработки данных из предоставляемых системой алгоритмов (Batch norm, Scalling, Gaussian Blur, VGG16);
5. Программа должна предоставлять пользователю возможность выбора оптимизируемого ML-алгоритма из предоставляемых системой ML-алгоритмов (Байесовская классификация, Логистическая регрессия, KNN, Модели ансамблей);
6. Программа должна предоставлять пользователю выбирать метод кросс-валидации из предоставляемых системой методов (K-Fold, Repeated Random Sub-sampling, Leave-one-out, Holdout, Resubstitution);
7. Программа должна отображать пользователю метрики классификации из предоставляемых системой метрик (error, accuracy, recall, specificity, precision, fall-out, F-score, Cohen’s Kappa, Микро и Макро усреднение по метрикам) по выбору пользователя;
8. Программа должна отображать пользователю графики обучения из предоставляемых системой графиков (AUC ROC, PR ROC) алгоритма по выбору пользователя;
9. Программа должна предоставлять пользователю возможность доразметки изображений\*;

\* – архив формата zip в виде набора папок, одна из которых имеет название "unlabeled", а имена остальных папок соответствуют названиям классов. Внутри папок находятся изображения в формате jpg, имена которых могут быть произвольными.

\*\* – алгоритм выбора данных для оптимальной доразметки предоставляет пользователю набор изображений, в котором пользователю необходимо отметить изображения указанного алгоритмом класса.

# Выбор модели жизненного цикла

В данной работе рассматривается задача разработки системы доразметки данных, которая включает в себя создание пользовательского интерфейса, интеграцию с алгоритмами классификации изображений, управление данными и обеспечение гибкости и высокой реактивности процесса доразметки.

В качестве модели жизненного цикла разработки системы была выбрана Agille-модель - это философия и методология разработки программного обеспечения, ориентированная на гибкость, итеративность и взаимодействие с заказчиком. Основное внимание уделяется созданию ценности для заказчика и способности быстро адаптироваться к изменяющимся требованиям. При выборе модели учитывались следующие цели разрабатываемой системы:

* **Гибкость и итеративность**: Разработка системы доразметки данных может потребовать постоянной адаптации к изменяющимся требованиям и условиям. Agile методологии предоставляют инструменты для итеративной разработки, что позволяет команде быстро реагировать на новые требования и вносить изменения в систему.
* **Визуализация рабочего процесса**: Визуализация может быть полезной для отслеживания процесса доразметки данных. Вы можете использовать Kanban-доску для отображения задач, связанных с разработкой системы, и их текущего статуса. Это поможет команде лучше видеть, где находятся задачи и какие этапы следующие.
* **Регулярные встречи (каденции)**: Для команды, разрабатывающей систему доразметки данных, регулярные встречи могут быть ключевым моментом для обсуждения прогресса, выявления проблем и планирования следующих шагов. Еженедельные встречи могут помочь участникам команды оставаться в курсе текущих задач и проблем.
* **Метрики и улучшение**: Важно использовать метрики для оценки производительности системы доразметки данных. Это может включать в себя измерение времени, необходимого для доразметки каждого изображения, пропускной способности системы, а также качества доразмеченных данных. Метрики помогут команде определить, где есть проблемы и как их улучшить.

На основе особенностей задачи и требований к проекту, выбор модели Agile может быть обоснован следующим образом:

* Scrum предоставляет фреймворк для управления проектом и продуктом, основанный на итеративных циклах разработки, называемых "спринтами". Это подходит для проектов, где есть необходимость часто обновлять итерфейс и функциональность системы в ответ на изменяющиеся требования пользователей. Scrum также способствует вовлечению заказчика и команды разработчиков в планирование и управление проектом.
* Kanban - это модель управления задачами, которая подходит для проектов с переменным потоком работы, где уделяется внимание минимизации задержек и оптимизации процессов. Этот подход может быть полезен в контексте разработки системы доразметки данных, где задачи постоянно меняются в зависимости от потребностей пользователей.

Таким образом, Agile-подходы обладают всеми требованиями, предъявляемыми к разработке системы, а также могут помочь команде разработчиков эффективно разрабатывать и улучшать систему доразметки данных, учитывая изменяющиеся потребности и обеспечивая высокое качество доразмеченных данных в задачах классификации изображений.

# Диаграмма прецедентов

Диаграмма прецедентов для системы оптимальной доразметки данных в задачах классификации изображений:

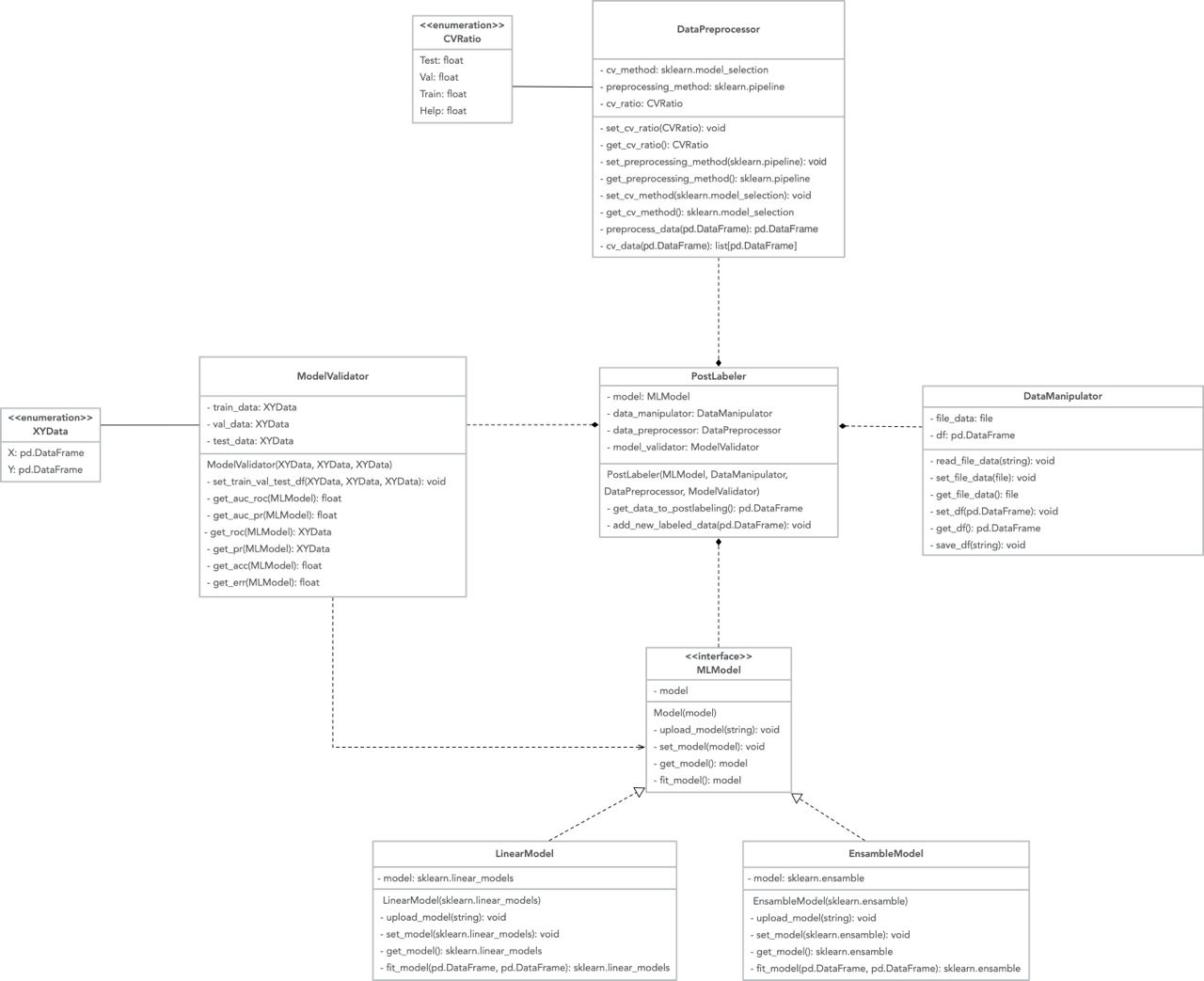


Так как разрабатываемое приложение является десктопным, пользователи не разделяются на роли. Из действий, доступных пользователю можно выделить:

* загрузку датасета изображений, который необходимо доразметить;
* выбор таких параметров, как:
  + ML-алгоритм, используемый для оптимизации доразметки,
  + алгоритм предобработки данных, который будет применен ко всем изображениям в датасете для выделения признаков, на которых будет обучаться ML-модель,
  + метод кросс-валидации;
* доразметку, в процессе которой можно:
  + выбрать изображения для доразметки,
  + просмотреть выбранные для доразметки изображения,
  + отменить выбор изображения;
* проверку качества разметки:
  + просмотр метрик классификации,
  + просмотр графика обучения ML-модели, в зависимости от количества размеченных данных;
* выгрузку обученной ML-модели.

# Диаграмма состояний

# Диаграмма классов



# Сценарии работы

Use-case сценарии:

1. Загрузка архива данных (датасета с частично-размеченными изображениями)

* Название: Загрузка архива данных в систему
* Краткое описание: Процесс загрузки и проверки архива данных.
* Участники: Пользователь.
* Предусловия: Пользователь запустил систему.
* Триггер: Пользователь выбирает и загружает архив формата zip с данными.
* Базовый сценарий:

1. Пользователь выбирает архив формата zip с данными на своем ПК.
2. Система проверяет формат архива, убеждаясь, что это действительно архив формата zip.
3. Система также проверяет наличие папки с именем "unlabeled" в архиве и наличие подпапок с именами, соответствующими классам.
4. Если данные соответствуют требованиям, они успешно загружаются в систему.

* Альтернативные сценарии:

1. Если формат или структура архива не соответствуют требованиям, выводится сообщение об ошибке "DATA\_FORMAT\_ERROR".
2. Если в загруженном архиве отсутствует папка с именем "unlabeled", выводится сообщение об ошибке "DATA\_UNLABELED\_NOT\_FOUND".

* Постусловие: Пользователь получает доступ к функционалу выбора алгоритма предобработки данных и просмотра загруженных изображений.

1. Просмотр изображений

* Название: Просмотр изображений.
* Краткое описание: Возможность просмотра изображений внутри папок архива с использованием фреймворка Matplotlib.
* Участники: Пользователь.
* Предусловия: Пользователь получил доступ к функционалу просмотра загруженных изображений.
* Триггер: Пользователь выбирает папку с изображениями для просмотра.
* Базовый сценарий:
  1. Пользователь выбирает папку с изображениями для просмотра.
  2. Система использует фреймворк Matplotlib для отображения группы изображений пользователю.
  3. Пользователь может просматривать изображения в интерфейсе Matplotlib.
* Альтернативные сценарии:

1. Если интерфейс отображения изображений не работает должным образом или происходит сбой, система предоставляет сообщение об ошибке.

* Постусловие: Пользователь ознакомился с изображениями в датасете.

1. Выбор алгоритма предобработки данных

* Название: Выбор алгоритма предобработки данных.
* Краткое описание: Процесс выбора пользователем определенного алгоритма предобработки данных из предоставленных в системе вариантов (например, BatchNorm, Scaling).
* Участники: Пользователь.
* Предусловия: Пользователь имеет доступ к функционалу выбора алгоритма предобработки данных.
* Триггер: Пользователь инициирует процесс выбора алгоритма предобработки данных.
* Базовый сценарий:
  1. Система предоставляет список доступных алгоритмов (например, BatchNorm, Scaling).
  2. Пользователь выбирает один из алгоритмов предобработки данных.
  3. Система сохраняет выбор пользователя для последующего использования.
* Альтернативные сценарии:

1. Если пользователь не выбрал алгоритм предобработки данных, то выводится предупреждение о том, что изображения будут преобразованы в массив чисел яркостей пикселей.

* Постусловие: Пользователь получает доступ к функционалу выбора ML-алгоритма.

1. Выбор ML-алгоритма

* Название: Выбор ML-алгоритма.
* Краткое описание: Процесс выбора пользователем определенного алгоритма машинного обучения из предоставленных в системе вариантов (например, Logistic regression, Naive Bayes, SVM, Random forest).
* Участники: Пользователь.
* Предусловия: Пользователь имеет доступ к функционалу выбора ML-алгоритма.
* Триггер: Пользователь инициирует процесс выбора ML-алгоритма.
* Базовый сценарий:
  1. Система предоставляет список доступных ML-алгоритмов (например, Logistic regression, Naive Bayes, SVM, Random forest).
  2. Пользователь выбирает один из ML-алгоритмов.
  3. Система сохраняет выбор пользователя для последующего использования.
* Альтернативные сценарии:

1. Если пользователь не выбрал ML-алгоритм, то выводится предупреждение о том, что будет использован заданный в системе ML-алгоритм по умолчанию (LogisticRegression).

* Постусловие: Пользователь получает доступ к функционалу выбора метода кросс-валидации.

1. Выбор метода кросс-валидации

* Название: Выбор метода кросс-валидации.
* Краткое описание: Процесс выбора пользователем определенного метода кросс-валидации из предоставленных в системе вариантов (например, K-fold, One-leave-out, Monte-Carlo).
* Участники: Пользователь.
* Предусловия: Пользователь имеет доступ к функционалу выбора метода кросс-валидации.
* Триггер: Пользователь инициирует процесс выбора метода кросс-валидации.
* Базовый сценарий:
  1. Система предоставляет список доступных методов кросс-валидации (например, K-fold, One-leave-out, Monte-Carlo).
  2. Пользователь выбирает один из методов кросс-валидации.
  3. Система сохраняет выбор пользователя для последующего использования.
* Альтернативные сценарии:

1. Если пользователь не выбрал метод кросс-валидации, то выводится предупреждение о том, что будет использован заданный в системе метод кросс-валидации по умолчанию (Monte-Carlo).

* Постусловие: Пользователю предоставляется функционал доразметки или разметки изображений.

1. Разметка или доразметка изображений
   1. Разметка изображений

* Название: Разметка изображений.
* Краткое описание: Возможность пользователя просматривать и размечать изображения: пользователь выбирает какой класс он будет размечать и выбирает изображения этого класса, при этом в отдельной области прописывается информация о требуемом минимальном количестве изображений в каждом классе и о количестве уже размеченных изображений каждого класса.
* Участники: Пользователь.
* Предусловия: Пользователю предоставлен функционал разметки изображений.
* Триггер: В какой-либо из папок-классов содержится меньше изображений, чем минимальное требуемое количество.
* Базовый сценарий:
  1. Пользователь выбирает класс, из предоставляемых системой (основываясь на именах папок-классов, содержащихся в архиве), который он будет размечать.
  2. Пользователь выбирает изображения выбранного класса.
  3. Система предоставляет пользователю возможность просматривать выбранные изображения, группируя их по классам.
  4. Система предоставляет пользователю возможность отмены разметки отдельных изображений.
  5. Система сохраняет изменения, внесенные пользователем.
  6. Пользователь может выбрать другой класс и повторить сценарий.
  7. Система выводит информацию о количестве размеченных данных в каждом классе, которая обновляется автоматически.
* Альтернативные сценарии:

1. Если количество размеченных изображений меньше минимального требуемого количества, то кнопка завершения является неактивной.

* Постусловие: Пользователю предоставляется функционал доразметки.
  1. Доразметка изображений
* Название: Доразметка изображений.
* Краткое описание: Возможность пользователя просматривать и доразмечать изображения в отдельном окне, выбирая их для доразметки и просматривая их в отдельной области этого окна.
* Участники: Пользователь.
* Предусловия: Пользователю предоставлен функционал доразметки изображений и количество размеченных изображений в каждой папке-классе не меньше заданного в системе минимального количества.
* Триггер: Пользователь запускает функционал доразметки изображений.
* Базовый сценарий:

1. Пользователь выбирает изображения из доступных для доразметки.
2. Система предоставляет пользователю возможность просматривать выбранные изображения в отдельной области.
3. Система предоставляет пользователю возможность отмены выбора отдельных изображений.
4. Система сохраняет изменения, внесенные пользователем.
5. Система обновляет график (находящийся в базовом окне) зависимости точности ML-модели от количества размеченных данных.

* Альтернативные сценарии:

1. Если изображения для доразметки отсутствуют, то система выводит сообщение о том, что все данные размечены.

* Постусловие: Пользователю предоставляется функционал выгрузки ML-модели и просмотра метрик и графиков классификации. Система загружает выполнение алгоритма выбора оптимальных изображений для последующей доразметки. Пользователь просматривает изменения в графике зависимости точности ML-модели от количества обученных данных: если точность его устраивает, то он переходит к этапу выгрузки ML-модели, иначе он может повторить данный сценарий.

1. Визуальное отображение метрик и графиков

* Название: Визуальное отображение метрик и графиков.
* Краткое описание: Отображение метрик классификации и графиков обучения пользователю.
* Участники: Пользователь.
* Предусловия: ML-модель обучена.
* Триггер: Пользователь запросил отображение метрик и графиков.
* Базовый сценарий:
  1. Пользователь запрашивает отображение метрик и графиков для оценки производительности ML-модели.
  2. Система генерирует и отображает метрики классификации (Accuracy, Precision, количество размеченных изображений каждого класса) и графики обучения (Accuracy от количества размеченных изображений, ROC от количества размеченных изображений, AUC от количества размеченных изображений) на экране.
  3. Пользователь анализирует результаты и может принимать решения о доразметке данных для улучшения метрик классификатора.
* Постусловие: Пользователь видит визуализированные метрики и графики, связанные с обучением ML-модели.

1. Сохранение ML-модели

* Название: Сохранение ML-модели.
* Краткое описание: Процесс сохранения ML-модели в формате ONNX пользователем.
* Участники: Пользователь.
* Предусловия: Метрики классификации соответствуют требованиям пользователя.
* Триггер: Пользователь выбрал функцию сохранения ML-модели.
* Базовый сценарий:
  1. Пользователь выбирает папку на жестком диске и вводит имя файла для ML-модели.
  2. Система сохраняет ML-модель в выбранную папку с выбранным именем файла в формате ONNX.
* Альтернативные сценарии:

1. Если путь или имя файла, выбранные пользователем, недопустимы или не существуют, система сообщает об ошибке и предлагает ввести корректные данные.

* Постусловие: ML-модель успешно сохранена по указанному пользователем пути и имени файла.

# Диаграммы последовательностей

# Краткое руководство пользователя

# План тестирования

# Тест-кейсы

# План интеграции

# Сопровождение ПО