TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и робототехники
Направление подготовки – 15.04.04 Автоматизация технологических процессов и производств
Отделение школы (НОЦ) – Отделение автоматизации и робототехники

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Детектирование и классификация дефектов в металлических заготовках с использованием методов машинного обучения

УДК 004.85:621.7.002.63

Студент

Группа	ФИО	Подпись	Дата
8ТМ81	Маляров Дмитрий Владимирович		02.06.2020

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Евгений Иванович	к.т.н., доцент		02.06.2020

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШПИБ	Конотопский Владимир Юрьевич	к.э.н		02.06.2020

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШПИБ	Горбенко Михаил Владимирович	к.т.н		02.06.2020

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТ	Ефимов Семен Викторович	к.т.н		02.06.2020

Томск – 2020 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код рез-та	Результат обучения (выпускник должен быть готов)	Требования ФГОС, критериев и/или заинтересованных сторон
Профессиональные		
Р1	применять глубокие естественно-научные, математические знания в области анализа, синтеза и проектирования для решения научных и инженерных задач производства и эксплуатации автоматизированных систем, включая подсистемы управления и их программное обеспечение.	Требования ФГОС (ПК-1, ПК-3, ОПК-1, ОПК-4, ОК-1, ОК-9), Критерий 5 АИОР (п. 1.1), согласованный с требованиями международных стандартов *EUR-ACE* и *FEANI*
Р2	воспринимать, обрабатывать, анализировать и обобщать научно-техническую информацию, передовой отечественный и зарубежный опыт в области теории, проектирования, производства и эксплуатации автоматизированных систем, принимать участие в командах по разработке и эксплуатации таких устройств и подсистем.	Требования ФГОС (ПК-3, ПК-4, ПК-7, ОПК-1, ОПК-3, ОК-1, ОК-4, ОК-5, ОК-6, ОК-9), Критерий 5 АИОР(пп. 1.1, 1.2), согласованный с требованиями международных стандартов *EUR-ACE* и *FEANI*
Р3	применять и интегрировать полученные знания для решения инженерных задач при разработке, производстве и эксплуатации современных автоматизированных систем и подсистем (в том числе интеллектуальных) с использованием технологий машинного обучения, современных инструментальных и программных средств.	Требования ФГОС (ПК-2, ПК-3, ПК-4, ПК-5, ПК-15, ПК-18, ОПК-3, ОПК-6, ОК-1, ОК-5, ОК-6, ОК-7), Критерий 5 АИОР (пп. 1.2), согласованный с требованиями международных стандартов *EUR-ACE* и *FEANI*
Р4	определять, систематизировать и получать необходимую информацию в области проектирования, производства, исследований и эксплуатации автоматизированных систем, устройств и подсистем.	Требования ФГОС (ПК-7, ПК-10, ПК-11, ПК-12, ПК-18, ОПК-4, ОПК-6, ОК-1, ОК-4, ОК-6, ОК-8), Критерий 5АИОР (п.1.3), согласованный с требованиями международных стандартов EUR-ACE и FEANI
Р5	планировать и проводить аналитические, имитационные и экспериментальные исследования для целей проектирования, производства и эксплуатации систем управления технологическим процессом и подсистем (в том числе интеллектуальных) с использованием передового отечественного и зарубежного опыта, уметь критически оценивать полученные теоретические и экспериментальные данные и делать выводы.	Требования ФГОС (ПК-1, ПК-2, ПК-3, ПК-4, ПК-5, ПК-6, ПК-13, ПК-17, ПК-18, ОПК-2, ОПК-3, ОК-1, ОК-3, ОК-4, ОК-6, ОК-7, ОК-8, ОК-9), Критерий 5АИОР (п. 1.4), согласованный с требованиями международных стандартов EUR-ACE и FEANI
Р6	понимать используемые современные методы, алгоритмы, модели и технические решения в автоматизированных системах и знать области их применения, в том числе в составе безлюдного производства.	Требования ФГОС (ПК-1, ПК-2 ПК-3, ПК-7, ОПК-1, ОПК-3, ОПК-4, ОК-5, ОК-9, ОК-10), Критерий 5 АИОР (п. 2.1), согласованный с требованиями международных стандартов EUR-ACE и FEANI

Универсальные		
Р7	эффективно работать в профессиональной деятельности индивидуально и в качестве члена команды.	Требования ФГОС (ПК-1, ПК-2 ПК-7, ПК-8, ПК-16, ПК-17, ОК-1, ОК-2, ОК-4, ОК-6, ОК-9), Критерий 5АИОР (п. 2.1), согласованный с требованиями международных стандартов *EUR-ACE* и *FEANI*
Р8	владеть иностранным языком на уровне, позволяющем работать в интернациональной среде с пониманием культурных, языковых и социально-экономических различий	Требования ФГОС (ПК-4, ПК-8, ПК-9, ПК-16, ОПК-4, ОК-5), Критерий 5 АИОР (п. 2.2), согласованный с требованиями международных стандартов *EUR-ACE* и *FEANI*
Р9	проявлять широкую эрудицию, в том числе знание и понимание современных общественных и политических проблем, демонстрировать понимание вопросов безопасности и охраны здоровья сотрудников, юридических аспектов, ответственности за инженерную деятельность, влияния инженерных решений на социальный контекст и окружающую среду	Требования ФГОС (ПК-5, ПК-8, ПК-15, ПК-16, ПК-18,ОПК-1, ОПК-4, ОПК-5, ОК-3, ОК-4, ОК-5, ОК-6, ОК-8, ОК-9), Критерий 5 АИОР (пп. 1.6, 2.3,), согласованный с требованиями международных стандартов *EUR-ACE* и *FEAN*
Р10	следовать кодексу профессиональной этики и ответственности и международным нормам инженерной деятельности	Требования ФГОС (ПК-8, ПК-11, ПК-16, ОПК-3, ОПК-6, ОК-4), Критерий 5 АИОР (пп. 2.4, 2.5), согласованный с требованиями международных стандартов *EUR-ACE* и *FEANI*
Р11	понимать необходимость и уметь самостоятельно учиться и повышать квалификацию в течение всего периода профессиональной деятельности.	Требования ФГОС (ПК-4, ПК-8, ОПК-3, ОПК-4, ОК-5, ОК-6, ОК-7, ОК-8), Критерий 5 АИОР (2.6), согласованный с требованиями международных стандартов *EUR-ACE* и *FEANI.*

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и робототехники
Направление подготовки (специальность) – 15.04.04 Автоматизация технологических процессов и производств
Отделение школы (НОЦ) – Отделение автоматизации и робототехники

УТВЕРЖДАЮ:
Руководитель ООП
_____ 02.06.20 Ефимов С.В.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

магистерской диссертации

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ТМ81	Маляров Дмитрий Владимирович

Тема работы:

Детектирование и классификация дефектов в металлических заготовках с использованием методов машинного обучения	
Утверждена приказом директора (дата, номер)	13.05.2020, 134-22/с

Срок сдачи студентом выполненной работы:	02.06.2020

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе *(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).*	1. Техническая информация об АСУТП разливки металлических заготовок заготовок; 2. Описание технологического процесса; 3. Выборка значений технологических параметров за определенный период.

Перечень подлежащих исследованию, проектированию и разработке вопросов *(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).*	1. Аналитический обзор принципов использования машинного обучения в задачах классификации и детектирования; 2. Анализ методов обработки информации для возможности работы с набором данных; 3. Анализ языков программирования, используемых для реализации методов машинного обучения; 4. Исследование математического описания методов; 5. Обучение моделей машинного обучения; 6. Подбор гиперпараметров моделей; 7. Тестирование моделей и анализ результатов; 8. Выполнение предпроектного анализа, планирование управления научно-техническим проектом, расчет бюджета исследования; 9. Анализ производственной безопасности, экологической безопасности, а также безопасности в чрезвычайных ситуациях. Исследование правовых и организационных вопросов обеспечения безопасности при проведении исследований.
Перечень графического материала *(с точным указанием обязательных чертежей)*	1. Схема автоматизации МНЛЗ; 2. Структурно-логическая схема оптической системы контроля; 3. Код моделей в Pytthon; 4. Диаграмма Ганта; 5. Диаграмммы распределения классов в наборе данных

Консультанты по разделам выпускной квалификационной работы
(с указанием разделов)

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Конотопский Владимир Юрьевич
Социальная ответственность	Горбенко Михаил Владимирович
Раздел на иностранном языке	Пичугова Инна Леонидовна

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Описание технологического процесса
Стек технологий
Подготовка исходных данных

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	24.02.2020

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Евгений Иванович	к.т.н., доцент		24.02.2020

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ТМ81	Маляров Дмитрий Владимирович		24.02.2020

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа – Инженерная школа информационных технологий и робототехники
Направление подготовки (специальность) – 15.04.04 Автоматизация технологических процессов и производств
Уровень образования – магистратура
Отделение школы (НОЦ) – Отделение автоматизации и робототехники
Период выполнения – осенний / весенний семестр 2019 /2020 учебного года

Форма представления работы:

магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	02.06.20

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
24.05.20	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	20
04.06.20	Социальная ответственность	20
01.06.20	Раздел на иностранном языке	60

СОСТАВИЛ:
Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Евгений Иванович	к.т.н., доцент		24.02.2020

СОГЛАСОВАНО:
Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Ефимов Семен Викторович	к.т.н.		24.02.2020

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8ТМ81	Маляров Дмитрий Владимирович

Школа		Отделение школы (НОЦ)	
Уровень образования	Магистратура	**Направление**	15.04.04 Автоматизация технологических процессов и производств

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:	
1. *Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих*	Использовать действующие ценники и договорные цены на потребленные материальные и информационные ресурсы, а также указанную в МУ величину тарифа на эл. энергию
2. *Нормы и нормативы расходования ресурсов*	—
3. *Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования*	Действующие ставки единого социального налога и НДС, ставка дисконтирования = 0,1 (см. МУ)
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. *Оценка коммерческого и инновационного потенциала НТИ*	Дать характеристику существующих и потенциальных потребителей (покупателей) результатов ВКР.
2. *Разработка устава научно-технического проекта*	Разработать проект такого устава в случае, если для реализации результатов ВКР необходимо создание отдельной организации или отдельного структурного подразделения внутри существующей организации
3. *Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок*	Построение плана-графика выполнения ВКР, составление соответствующей сметы затрат, расчет цены результата ВКР.
4. *Определение ресурсной, финансовой, экономической эффективности*	Оценка экономической эффективности использования результатов ВКР, характеристика других видов эффекта
Перечень графического материала *(с точным указанием обязательных чертежей):*	
1. *«Портрет» потребителя результатов НТИ* 2. *Сегментирование рынка* 3. *Оценка конкурентоспособности технических решений* 4. *Диаграмма FAST* 5. *Матрица SWOT* 6. *График проведения и бюджет НТИ - выполнить* 7. *Оценка ресурсной, финансовой и экономической эффективности НТИ - выполнить* 8. *Потенциальные риски*	

Дата выдачи задания для раздела по линейному графику	24.02.2020

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Конотопский В.Ю.	к.э.н.		24.02.2020

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ТМ81	Маляров Дмитрий Владимирович		24.02.2020

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8ТМ81	Маляров Дмитрий Владимирович

Школа		Отделение (НОЦ)	
Уровень образования	Магистратура	Направление	15.04.04 Автоматизация технологических процессов и производств

Тема ВКР:

Детектирование и классификация дефектов в металлических заготовках с использованием методов машинного обучения	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является программный модуль – совокупность моделей для автоматической идентификации и классификации дефектов конечной продукции, в основе алгоритма работы которых используются методы машинного обучения. Данный программный модуль может использоваться в технологической деятельности промышленных предприятий, лабораторий и т.д, нуждающихся в контроле качества продукции.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	Рассмотрены специальные правовые нормы трудового законодательства и организационные мероприятия при компоновке рабочей зоны. Требования к рабочему месту при выполнении работ сидя устанавливаются в ГОСТ 12.2.032-78; Гигиенические требования к ПЭВМ и организации работ регулируется СанПиНом 2.2.2/2.4.1340-03
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Выявлено воздействие на исследователя физических факторов, таких как, повышенный уровень электромагнитных полей, недостаточная освещенность рабочей зоны, повышенный уровень шума на рабочем месте, повышенная или пониженная влажность, опасность поражения электрическим током.
3. Экологическая безопасность:	Утилизация компьютерной техники и периферийных устройств

4. Безопасность в чрезвычайных ситуациях:	Наличие возможности возгорания в следствии короткого замыкания из-за ошибки персонала и нарушения целостности электрических проводов. Наиболее типичная ЧС - пожар. Основными мероприятиями по пожарной безопасности являются проведение пожарной профилактики, наличие противопожарного водопровода оборудования, а также средств оповещения.

Дата выдачи задания для раздела по линейному графику	24.02.2020

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШПИБ	Горбенко Михаил Владимирович	к.т.н		24.02.2020

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ТМ81	Маляров Дмитрий Владимирович		24.02.2020

Реферат

Пояснительная записка содержит 140 страниц машинописного текста, 15 таблиц, 52 рисунка, список использованных источников из 45 наименований и одного приложения.

Ключевые слова: машинное обучение, набор данных, метрика качества.

Цель работы: автоматизация процесса детектирования и классификации дефектов металлических заготовок с интеграцией в АСУТП для повышения эффективности контроля качества продукции.

В данной работе рассматривается технологический процесс непрерывного литья заготовок, процесс обработки и анализа данных, а также сравнение посредством различных метрик качества таких методов машинного обучения, как дерево решений, метод ближайших соседей, логистическая регрессия и другие.

Степень внедрения: стадия разработки проектной документации.

Область применения: металлургическая отрасль, цифровое производство.

Фактором экономического эффекта, достигаемым в предложенной работе, является возможность автоматизации процесса контроля качества продукции благодаря чему минимизируется время на принятие решений, а также влияние человеческого фактора при осуществлении оценки качества.

Для выполнения работы использовались программные продукты: язык программирования Python, среда разработки Jupyter Notebook, MS Excel для хранения набора данных, OPC-сервер “Process Simulator” для передачи значений параметров в ПЛК, среда разработки TIA Portal, в которой осуществляется симуляция работы ПЛК.

Оглавление

Обозначения и сокращения

В данной работе используются следующие термины аббревиатуры и сокращения:

АРМ – Автоматизированное рабочее место

АСУ – Автоматизированная система управления

АСУП – Автоматизированная система управления предприятием

АСУТП – Автоматизированная система управления технологическим процессом

ИМ – измерительный механизм

КИМ – Координатно-измерительная машина

МНЛЗ – Машина непрерывного литья заготовок

МО – Машинное обучение

ОК – Объект контроля

ОС – Операционная система

ПЛК – Программируемый логический контроллер

САР – Система автоматического регулирования

СИ – Средство измерения

СУБД – Система управления базой данных

ЦОП – Центральный операторский пункт

KNN – (англ. k Nearest Neighbor) Метод К-ближайших соседей

SVM – (англ. support vector machine) Метод опорных векторов

Термины и определения

В данной работе применены следующие термины с соответствующими определениями:

АСУ – комплекс аппаратных и программных средств, а также персонала, предназначенный для управления различными процессами в рамках технологического процесса, производства, предприятия

АСУП – комплекс программных, технических, информационных, лингвистических, организационно-технологических средств и действий квалифицированного персонала, предназначенный для решения задач планирования и управления различными видами деятельности предприятия.

АСУТП – человеко-машинный комплекс, обеспечивающий управление технологическими процессами на современных механизированных и автоматизированных промышленных предприятиях.

Выброс (в статистике) – результат измерения, выделяющийся из общей выборки.

Гиперпараметр – свойство алгоритма обучения, как правило имеющее числовое значение, не вычисляемое при этом самим алгоритмом.

Диаграмма рассеяния (также точечная диаграмма, англ. scatter plot) – математическая диаграмма, изображающая значения двух переменных в виде точек на декартовой плоскости.

Машинное обучение – раздел информатики, посвященный созданию алгоритмов, опирающихся на набор данных о каком-либо явлении.

Признак (англ. feature) объекта – это результат измерения некоторой характеристики объекта.

Процентиль – мера, в которой процентное значение общих значений равно этой мере или меньше ее

Стальковш – цилиндрический стальной ковш с футеровкой на внутренних стенках, используется для транспортировки жидкой стали.

СУБД – совокупность программных и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения многопользовательского доступа к данным.

Введение

Проверка контроля качества заключается в проверке всех видов значений (количественных, качественных и т.д.) процесса или продукции на соответствие техническим требованиям. Процесс контроля качества является неотъемлемой частью любого производственного процесса и позволяет оценить надежность изготовления продукции и будущего ее использования. Сущность такого контроля заключается в приобретении информации о состоянии объекта и сопоставлении полученных результатов с установленными требованиями, зафиксированными в стандартах, условиях поставки и т.д. Основной задачей контроля является отделение качественных изделий от бракованных. В современных условиях все больше предприятий сосредотачиваются на тщательном контролировании процесса и придерживаются концепции “менеджмента качества”, а не на детектировании брака. Качество продукции – один из важнейших показателей, поэтому выживаемость предприятия в рыночных условиях, повышение эффективности производства и другие важнейшие аспекты напрямую зависят от повышения контроля [1].

Если говорить о контроле качества в процессе литья заготовок, то механические свойства, параметры геометрии являются основными признаками объекта. Благодаря использованию АСУ производители сталелитейной продукции обеспечивают потребителям вышеупомянутые показатели.

Состояние поверхности продукта (сталелитейной заготовки) одна из основных составляющих качества продукта в целом. Как правило, контроль поверхности осуществляется на конечных этапах производства.

Для детектирования и классификации дефектов как правило используются визуальные способы: либо применяются специальные системы, либо внешний осмотр [3].

Проведение осмотра затрудняют такие факторы как высокая температура продукта, окисление поверхности, а также движение полосы

(конвейера), в результате анализ дефектов возможен только после остывания. Это приводит к увеличению количества бракованной продукции, особенно в случае прокатного происхождения дефекта [2].

Как следствие необходимо как с высокой точностью обеспечивать детектирование наличия дефекта, так и правильную его классификацию.

Проблематика

В настоящее время существуют системы контроля качества поверхности, способные получать изображения поверхности, определять ряд параметров дефектов и т.д. Однако не все системы могут осуществлять детектирование дефектов, а также определять принадлежность дефектов к определенному классу на основе их параметров.

В подобной ситуации вышеуказанные действия осуществляются группой экспертов.

Человеческий фактор при выполнении визуального контроля приводит к допуску бракованных продуктов в класс годных.

Кроме того, визуальный анализ характеризуется низкой производительностью. Отмеченные недостатки обуславливают актуальность задачи автоматизации операции обнаружения дефектных изделий [7].

Использование методов машинного обучения для детектирования и классификации дефектов позволяет минимизировать вышеуказанные негативные эффекты.

Обзор литературы

В ВКР [1] дано понятие контроля качества продукции, проанализированы важность и положительные эффекты от повышения качества продукции в условиях современного рынка.

В статье [2] описывается важность контроля качества непосредственно при производстве стальных заготовок. Рассматривается один из вариантов детектирования дефектов при помощи СККП, указывается ее состав и алгоритм работы.

В учебном пособии [3] собрана информация о видах дефектов сталей и сплавов, указаны дефекты поверхностей слитков и т.п. Проиллюстрирован внешний вид дефектов, их структуры, а также указаны причины появления дефектов и способы по их предупреждению. Идентификация дефектов облегчается благодаря описанию характерных признаков дефектов.

Учебное пособие [4] содержит основные сведения о видах заготовок, методах получения, их особенностях и использовании.

ВКР [5] дает подробное описание технологического процесса непрерывного литья заготовок с использованием машины МНЛЗ, приводит схему автоматизации, а также дает полное описание системы АСУТП.

Работа [10] демонстрирует практические аспекты применения алгоритмов машинного обучения, а также а дает описание и представление возможностей библиотек NumPy и Matplotlib.

В учебном пособии [20] дается формализованное описание задачи классификации, а также дано математическое обоснование использования различных методов (метод градиентного спуска и т.д) для оптимизации целевых функций моделей

Выпускная квалификационная работа [24] дает представление о методах SVM, KNN и других, используемых в задачах классификации. Также в работе имеется сравнительный анализ различных метрик качества тестируемых моделей с объяснением нюансов использования каждой из них.

Статья [25] дает полное описание метода “Логистическая регрессия”, а также предоставляет математическое описание видов регуляризации целевой функции.

В работе [26] рассмотрены математические основы и графическая демонстрация как классических алгоритмов, так и современных. Кроме того, описаны основные способы обработки исходного набора данных.

Работа [28] представляет широкий спектр основных понятий и методов машинного обучения таких как классические логистическая регрессия, метод ближайших соседей, так и современные методы, такие как метод опорных векторов, бустинг и т.д., кроме того, в книге продемонстрирована реализация некоторых алгоритмов на языке Python.

Работа [30] предоставляет описание практической реализации различных методов машинного обучения, а также описывает принципы работы современны методов (XGBoost и др.).

Цель

Целью выполнения магистерской диссертации является:

1) повышение точности и скорости детектирования и классификации дефектов поверхности сталелитейных заготовок;

2) определение единого набора методов для обработки и подготовки данных с технологического процесса;

3) выявление ключевых параметров, определяющих дефекты.

Задачи

Выполнение поставленных целей обуславливается решением следующих задач:

1) найти и определить исходный набор данных, необходимый для обучения;

2) сформировать тренировочный и тестовый наборы данных;

3) определить набор методов (модели), с помощью которых будет происходить обучение;

4) обучить модели на тренировочном наборе данных и сравнить результаты;

5) определить наилучшие параметры для каждой из моделей;

6) сравнить точность моделей на тестовом наборе данных;

7) определить стек технологий для интеграции методов МО в работу АСУТП.

1 Описание технологического процесса

Для определения взаимосвязи между исходным набором данных и технологическим процессом, а также для понимания состава и параметров данных, необходимо понимать основы протекания процесса литья заготовок, причины возникновения и виды возможных дефектов и т.д.

Процесс начинается с доставки расплавленной стали в стальковше к МНЛЗ и установки ее в позицию разливки.

Непрерывная разливка происходит на МНЛЗ и применяется в большинстве сталеплавильных процессах.

Типы МНЛЗ: радиальные, вертикальные и криволинейные. В настоящее время широко используются 1-ый и 3-ий тип, поскольку они способны формировать изогнутые заготовки (с определенным радиусом).

Технологический процесс начинается с кристаллизатора, где происходит начальное охлаждение заготовки, далее он попадает в канал вторичного охлаждения, представляющий собой совокупность роликовых секций. Заготовка при кристаллизации проходит четверть окружности.

После перехода в горизонтальное положение непрерывно литой слиток выпрямляется в правильно-тянущих клетях и разрезается на мерные заготовки.

Схема вертикальной МНЛЗ и система автоматики показана на рисунке 1. Сталь подается из сталеплавильного отделения в ковше, далее поступает в кристаллизатор. Заготовка с остывшими стенками вытягивается благодаря тянущей клети и при этом проходит зону вторичного охлаждения. Автоматическая газорезка делит слиток на мерные длины.

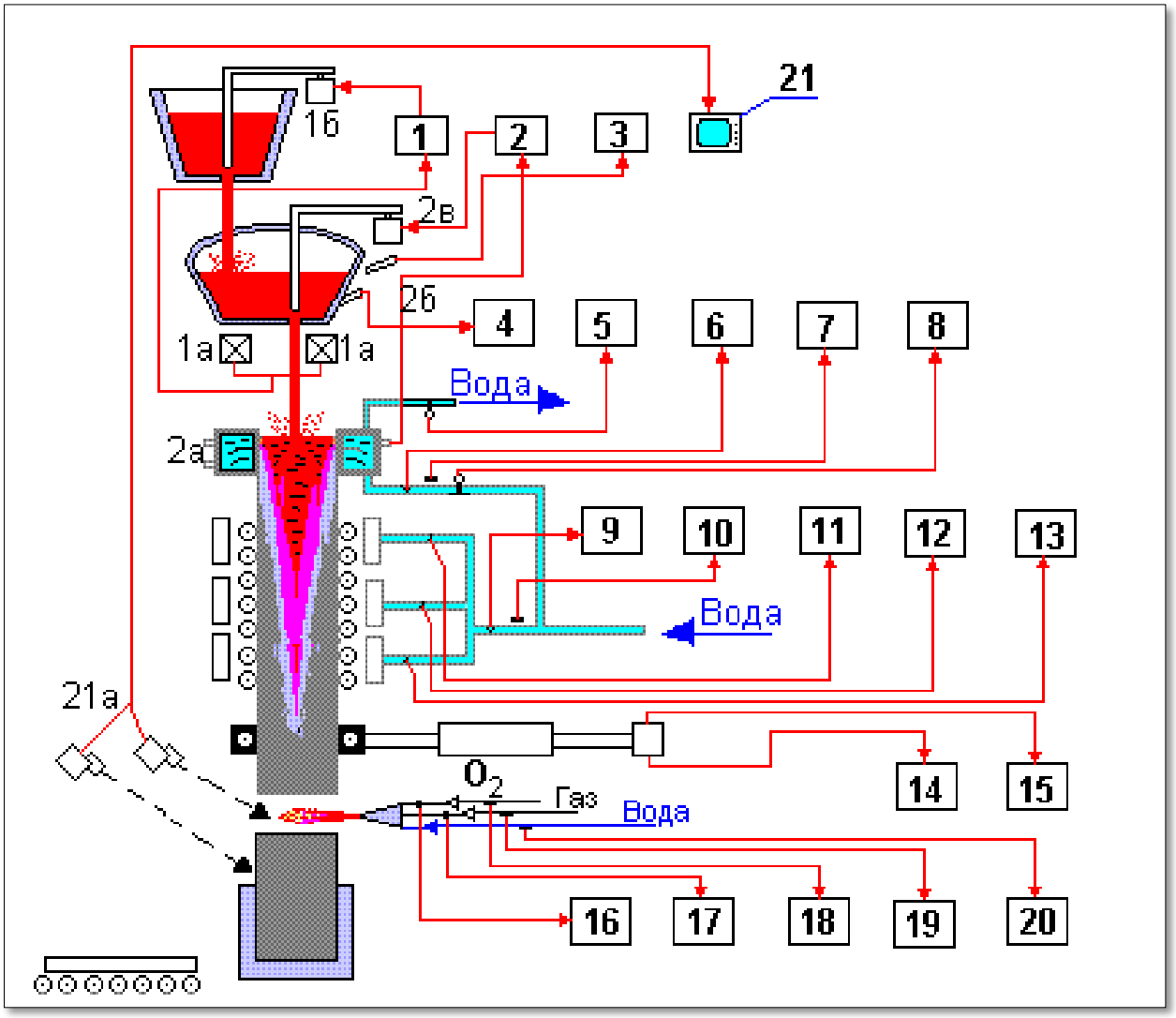


Рисунок 1 – Схема автоматизации МНЛЗ вертикального типа

Клеть, газорезку, выдачу слитков и т.д. в движение приводятся автоматизированными электроприводами. Контроль и управление элементами МНЛЗ осуществляется мнемосхемой и панелью аварийной сигнализации. Предупредительная и аварийная сигнализации предусмотрены для:

1) отключение механизма качения;
2) превышение времени резке заготовки;
3) останов клети;
4) отсутствие тележки под слитком и др.

Установки непрерывной разливки стали работают в стационарном режиме и требуют для поддержания такого режима совершенной системы автоматического контроля и регулирования. Отклонения от наилучшего

режима разливки, вызываемые различными возмущениями, могут приводить к уменьшению производительности, ухудшению качества металла и возникновению аварийных ситуаций. Системы автоматического контроля и регулирования МНЛЗ способствуют устранению возмущений и обеспечивают наиболее рациональный режим разливки и безопасную работу агрегата.

Системы контроля и регулирования процесса выполняют следующие функции:

а) регулирование уровня металла в кристаллизаторе и промежуточном ковше;

б) регулирование расхода воды в зоне вторичного охлаждения

в) контроль температурного состояния конструктивных элементов агрегата с целью устранения аварийных режимов;

г) автоматическая резка слитка на мерные длины, т.е. на заготовки заданной длины.

Уровень металла в промежуточном ковше (рис. 1) стабилизируется комплектом аппаратуры, состоящим из тензометрических датчиков массы 1а и регулирующего устройства 1, управляющего приводом стопора ковша 16.
В результате с помощью стабилизации массы метала в ковше регулируется его уровень. Регулирующее устройство работает по двухпозиционному закону регулирования.

Качественная кристаллизация слитка и безопасная работа установки во многом зависят от стабильного уровня металла в кристаллизаторе. Аварийный высокий уровень металла является причиной перелива стали, аварийный низкий уровень – к выходу метала из внутреннего объема слитка.

Контур регулирования состоит из уровнемера с источником гамма-излучения 2а и приемника 2б. Исполнительным механизмом является стопор, который опускается или поднимается при отклонении уровня металла, что обеспечивает изменение проходного сечения в днище ковша.

Температура стали в промежуточном ковше периодически контролируется термопарой погружения с регистрацией на потенциометре 3.

Для наблюдения за степенью прогрева кладки ковша перед наполнением металлом в ней устанавливают термопару с регистрирующим потенциометром 4.

При разливке слитков небольшого сечения перекрытие струи металла стопором может привести к ее деформации, разбрызгиванию металла по стенкам кристаллизатора и ухудшению условии образования качественного слитка. Поэтому применяется способ регулирования уровня металла в кристаллизаторе путем изменения скорости вытяжки слитка при неизменной подаче жидкого металла из промежуточного ковша. В этом случае регулятор 2 воздействует на привод тянущей клети.

Регулирование расхода воды по секциям системы вторичного охлаждения необходимо для организации правильного режима кристаллизации и охлаждения металла по высоте слитка и но его периметру. Равномерное охлаждение граней слитка устраняет возможную его деформацию из-за температурных напряжений. Расходы воды по секциям вторичного охлаждения контролируются стандартными комплектами 11, 12, 13 с измерительными диафрагмами или ротаметрами в качестве первичных приборов. Изменение расхода воды осуществляется дистанционным ручным управлением регулировочными клапанами на водопроводах.

Давление и расходы воды на кристаллизатор и вторичное охлаждение контролируются приборами 6, 7 и 9, 10, причем манометры 7 и 10 снабжены сигнальными контактами для сигнализации о недопустимом падении давления воды.

Контроль тепловой работы и температурного состояния кристаллизатора осуществляется измерением температуры воды на его выходе термометром сопротивления с электронным автоматическим мостом 5. Аналогичным комплектом 8 контролируется температура воды на входе в кристаллизатор. Отсчет общей длины слитка и мерных длин осуществляется с помощью датчиков импульсов, установленных на валу редуктора тянущей

клети и прибора 14, включающего в себя счетчики импульсов и показывающие индикаторы. Тахогенератором и прибором 15 определяется скорость движения металла.

Работа автоматической газорезки требует соответствующих количеств газа, кислорода и охлаждающей воды. Давление в подводящих линиях контролируется манометрическими комплектами с сигнальными контактами 18, 19, 20, а расходы газа и кислорода - измерительными диафрагмами с приборами 16 и 17.

Для наблюдения за работой отдельных узлов агрегата, например, за работой газорезки и механизма приема и выдачи отрезанных слитков, применяется промышленная телевизионная установка, состоящая из камер и приемника изображения 27. При исследовании и наладке контролируется температура слитка на различных участках при помощи пирометров излучения.

Контроль и управление МНЛЗ осуществляется операторами с пультов, расположенных в ЦОП, пульта газорезки и выдачи слитков. Дистанционный пуск и остановка МНЛЗ, регулирование скорости вытягивания заготовок, работа охлаждения, подача смазочных материалов и т.д. также осуществляются из ЦОП.

Для оперативной связи между пультами управления МНЛЗ и между установкой и другими участками цеха служит громкоговорящая связь.

Схема системы автоматизации МНЛЗ радиального типа приведена па рисунке 2.

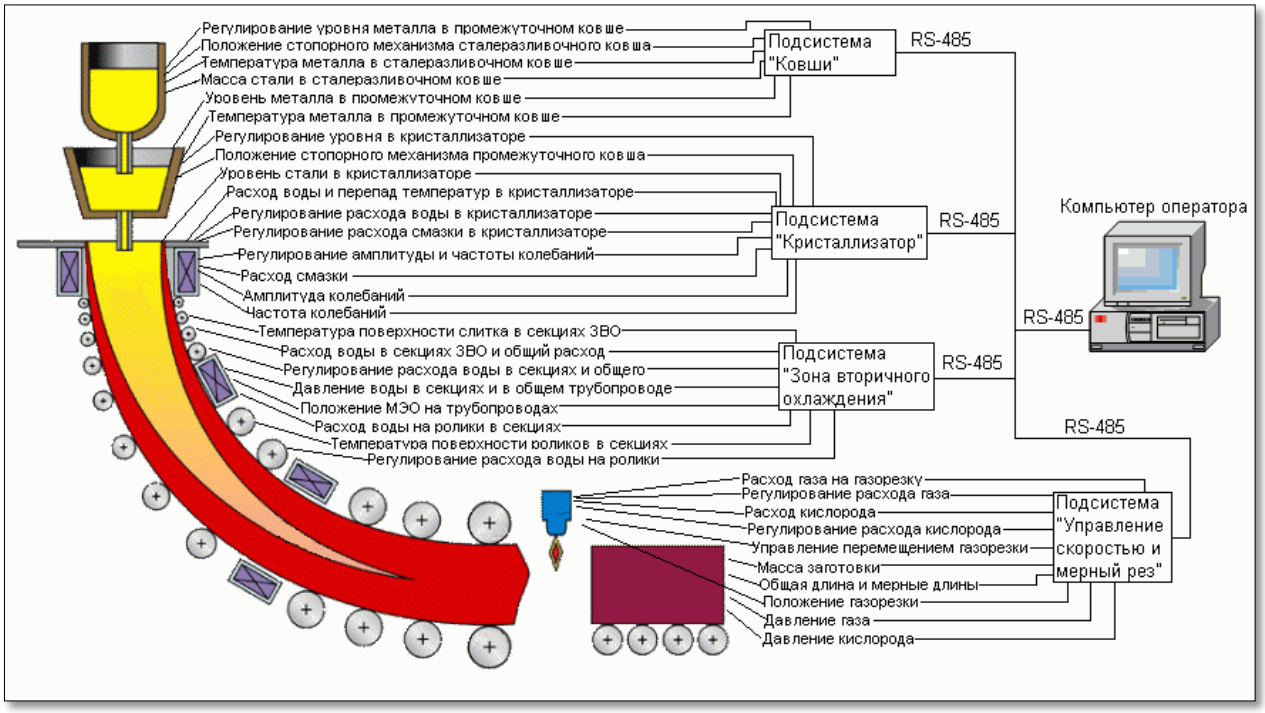


Рисунок 2 – Структурная схема автоматизированной системы управления МНЛЗ

Достоинством установок такого типа является то, что они требуют для своего сооружения цеха меньшей высоты из-за изгиба слитка и общие капитальные затраты на сооружение цехов при этом уменьшаются. Основные функции управления, средства контроля и узлы регулирования для радиальных МНЛЗ такие же, как и для вертикальных установок.

Уровень металла в промежуточном ковше и кристаллизаторе регулируется блоками 1 и 2, блоки 3 и 4 предназначены для регулирования охлаждения кристаллизатора и металла, блоки 5 и 6 регулируют расход газа и кислорода на газорезку.

Отличие радиальных МНЛЗ состоит также в том, что в них для управления сливом металла из разливочного ковша в промежуточный и из последнего в кристаллизатор применяются не стопорные устройства, а шиберные.

К локальным системам относятся:

– САР миксерного отделения сталеплавильного цеха;

– система регулирования уровня металла в промежуточном ковше;

– система регулирования уровня металла в кристаллизаторе;

– САР теплового режима кристаллизатора;

– САР теплового режима вторичного охлаждения и др.

1.1 АСУТП непрерывной разливки стали

АСУ ТП непрерывной разливки стали входит обычно как составная часть в интегрированную АСУ сталеплавильным, например, конвертерным цехом. В целом АСУ ТП должна обеспечивать за счет стабилизации и оптимизации технологических режимов разливки, повышение производительности; повышение выхода годного металла; уменьшение числа аварийных режимов работы и повышение работоспособности МНЛЗ, улучшение условий труда обслуживающего персонала.

Основные функции АСУ ТП непрерывной разливки стали:

1. Измерение параметров:

а) T стали в сталеразливочном ковше;

б) T стали в промежуточном ковше;

в) m стали в сталеразливочном ковше;

г) L металла в промежуточном ковше;

д) L металла в кристаллизаторе;

е) F вытягивания слитка из кристаллизатора;

ж) V вытягивания слитка (скорости разливки) ;

з) F и P воды на кристаллизатор;

и) T воды на кристаллизаторе;

к) F смазочных материалов в кристаллизатор;

л) F и P воды в зоне вторичного охлаждения;

м) T поверхности слитка;

н) общая и мерная l слитка.

2. Суммирование и подсчет:

а) Q и d оболочки слитка в зоне вторичного охлаждения;

б) основных параметров разливки (V разливки, F смазки, F воды на кристаллизатор и на вторичное охлаждение);

в) экономических показателей работы МНЛЗ.

Управление:

1. Регулирование:

а) L металла в промежуточном ковше:

б) L металла в кристаллизаторе;

в) F воды на кристаллизатор;

г) F воды на секции зоны вторичного охлаждения;

д) F технологической смазки:

е) F газа и кислорода на газорезку.

2. Управление процессом:

а) пусковой режим МНЛЗ;

б) режимом охлаждения;

в) привод тянущих клетей;

г) разрезка слитка на мерные длины;

Также осуществляются следующие функции:

1. сигнализацию основных технологических параметров;

2. накопление и хранение информации о режиме отливки и условиях формирования каждой заготовки;

3. регистрацию аварийных ситуаций в журнале событий;

1.2 Система контроля качества

Ручной контроль качества все еще превалирует на многих видах производств, так как любое промышленное предприятие имеет целый арсенал обученных сотрудников и отточенных стандартов качества, на соответствие которым изделия проходят проверку.

Один из самых распространенных методов контроля – оптический, в основе которого лежит взаимодействие оптического излучения и объекта

контроля [31]. В настоящее время автоматизированная оптическая система контроля (ОСК) качества способна в значительной степени сократить непосредственное участие работников в процессе проверки качества на всех видах производственных линий, отведя человеку роль руководителя процесса [6].

Структурно-логическая схема ОСК представлена на рисунке 3.

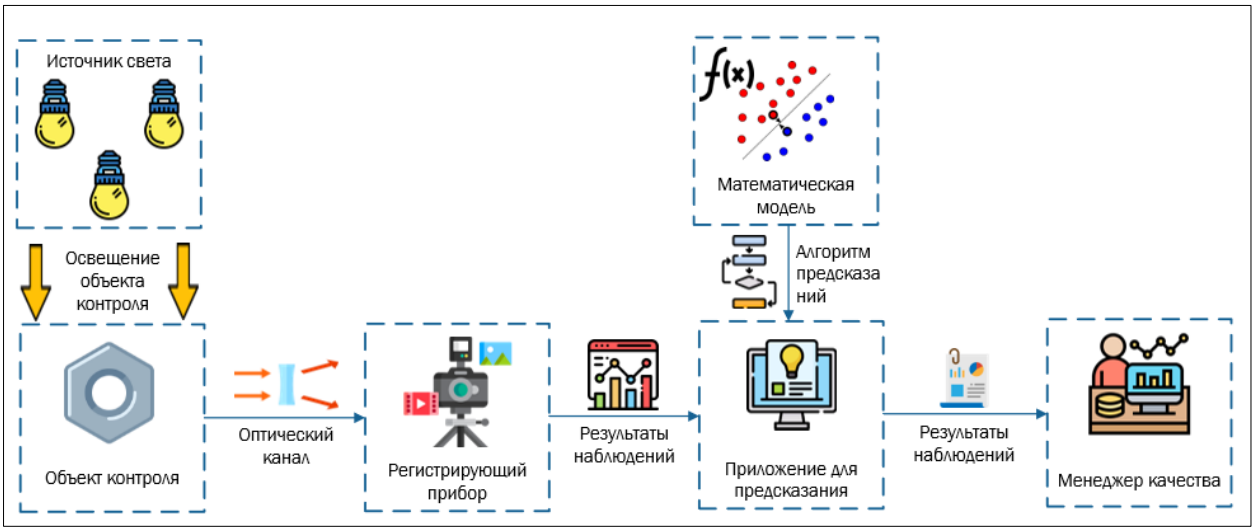


Рисунок 3 – Структурно-логическая схема ОСК

Процесс работы системы выглядит следующим образом. Генерируемое источником света оптическое излучение взаимодействует с объектом контроля, возникающий при этом сигнал поступает в средство измерения. Результаты измерений подвергаются процедуре оценивания со стороны предиктивного приложения (модели). По результатам работы будет сформирован отчет, содержащий информацию об отнесении контролируемого объекта в класс годных или бракованных.

Иерархия АСУТП ОСК состоит из следующих уровней:

1) 1-ый уровень – уровень ИМ и СИ:

– КИМ и лазерные сканеры для измерения параметров геометрии ОК;

– SMART-камеры с высоким разрешением для замера параметров поверхности ОК.

2) 2-ой уровень – контроллерный:

– Передача результатов измерений в ПЛК посредством интерфейса Ethernet;

– Хранение результатов измерений на сервере БД;

– Обработка данных и реализация методов МО с выдачей результата о наличии дефекта и его типе на сервере приложений.

3) 3-ий уровень – операторный:

– Управление технологическим процессом и мониторинг параметров оператором АСУТП посредством АРМ.

4) 4-ый уровень – уровень АСУП:

– Управление технологическим процессом со стороны менеджеров на основании информации о текущем состоянии качества продукции посредством АРМ.

Таким образом, структурная схема для вышеописанной 4-ех уровневой АСУТП имеет следующий вид:

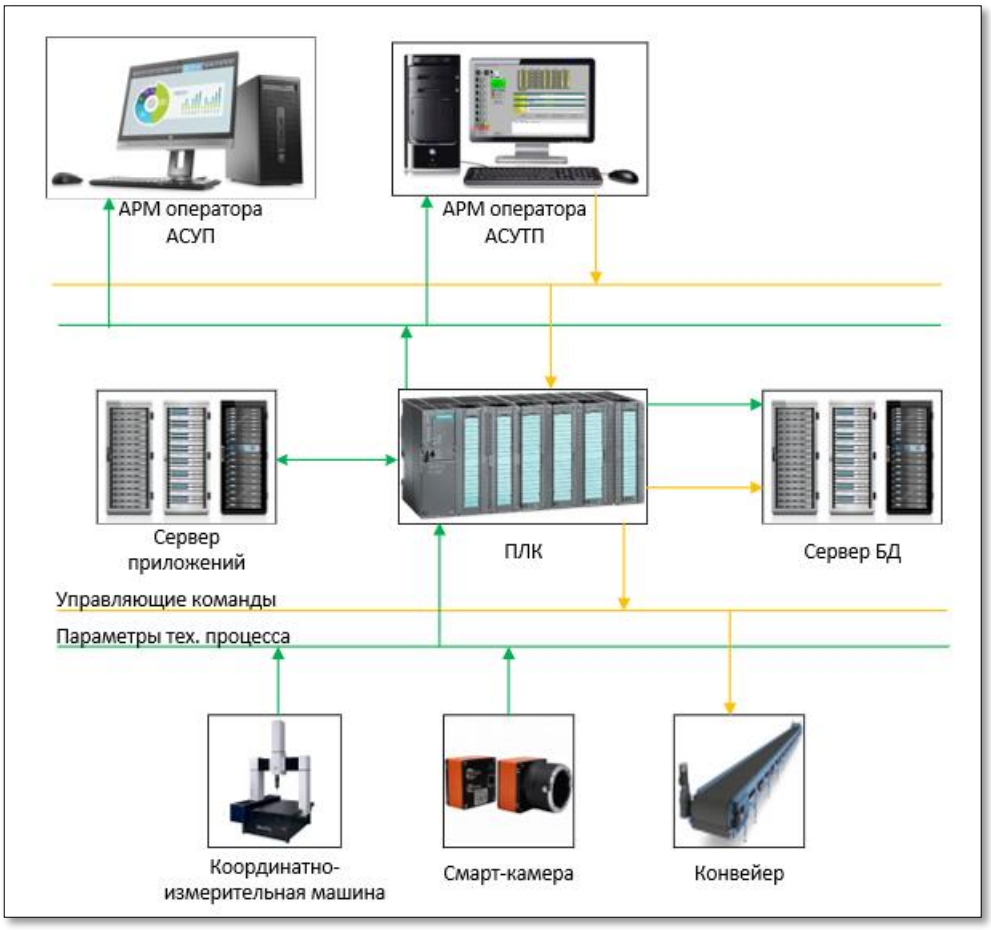


Рисунок 4 – Структурная схема

Подобная система может состоять из нескольких камер, в том числе инфракрасных, системы подсветки структурированным светом, серверов хранения и обработки данных с рабочими станциями.

Благодаря КИМ и смрат-камерам осуществляется снятие различных параметров дефектов заготовок, таких как изменение геометрии, координаты, количество пикселей, приходящихся на область дефекта и т.д.

Значения параметров фиксируются на сервере БД и также передаются на АРМ оператора АСУТП и АРМ.

Результаты детектирования и классификации дефектов передаются на пост управления, а также хранятся в БД, благодаря чему оператор стана обладает текущей информацией о состоянии поверхности.

2 Стек технологий

2.1 Выбор языка программирования

Анализ данных и создание моделей для использования методов машинного обучения возможно в нескольких языках программирования:

1) Python;

Представляет собой высокоуровневый язык, являющийся наиболее популярным инструментом анализа данных, в том числе и направлении “BigData”. Кроме того, обладает большим количеством библиотек, и может быть адаптирован к любой ОС.

Одним из недостатков является сложность отслеживания ошибок в коде.

2) R;

R также пользуется популярностью в анализе данных.

Он также открытым исходным кодом, имеет большое количество библиотек.

R поддерживает различные платформы, а также взаимодействие с другими языками программирования. Также R поддерживает визуализацию данных.

По сравнению с Python, имеет специфические особенности, затрудняющие его изучение (непривычные структуры данных, индексирование с единицы и т.д).

Несмотря на широкие возможности, R непопулярен как Python, в результате чего количество документации к нему и его библиотекам меньше.

3) C++.

C++ старее вышеупомянутых языков, однако может обеспечивать контроль выше многих языков программирования за счет обладания возможностями как низко- так и высокоуровневого языка программирования.

Из его широких возможностей следует его гибкость, подходящая для энергоемких программ, в том числе и МО задач.

В возможностях контроля языка C++ заключается один из главных минусов для неопытных пользователей – для создания новых приложений необходимо столкнуться с написанием сложного объемного кода, как следствие язык C++ сложен в овладении.

Анализируя преимущества и недостатки представленных языков, а также учитывая текущий уровень владения ими, в качества основного языка для обработки данных и построения моделей был выбран язык Python.

2.2 Среда разработки

При написании кода в Python, интегрирования модулей и библиотек для построения больших систем, текстового редактора недостаточно. Требуется интегрированная среда разработки (IDE).

IDE – это программа, предназначенная для разработки программного обеспечения. Как следует из названия, IDE объединяет несколько инструментов, специально предназначенных для разработки. Эти инструменты обычно включают редактор, предназначенный для работы с кодом (например, подсветка синтаксиса и автодополнение); инструменты сборки, выполнения и отладки; и определённую форму системы управления версиями [17].

Однако взаимодействие с данными предполагает одновременную работу с кодом, изображениями, графиками и т.д. Одним из лучших решений является Jupyter Notebook.

Jupyter Notebook – это мощный инструмент для разработки и представления проектов Data Science, Machine Learning и т.д в интерактивном виде. Он объединяет код и вывод все в виде одного документа, содержащего текст, математические уравнения и визуализации.

Такой пошаговый подход обеспечивает быстрый, последовательный процесс разработки, поскольку вывод для каждого блока показывается сразу же.

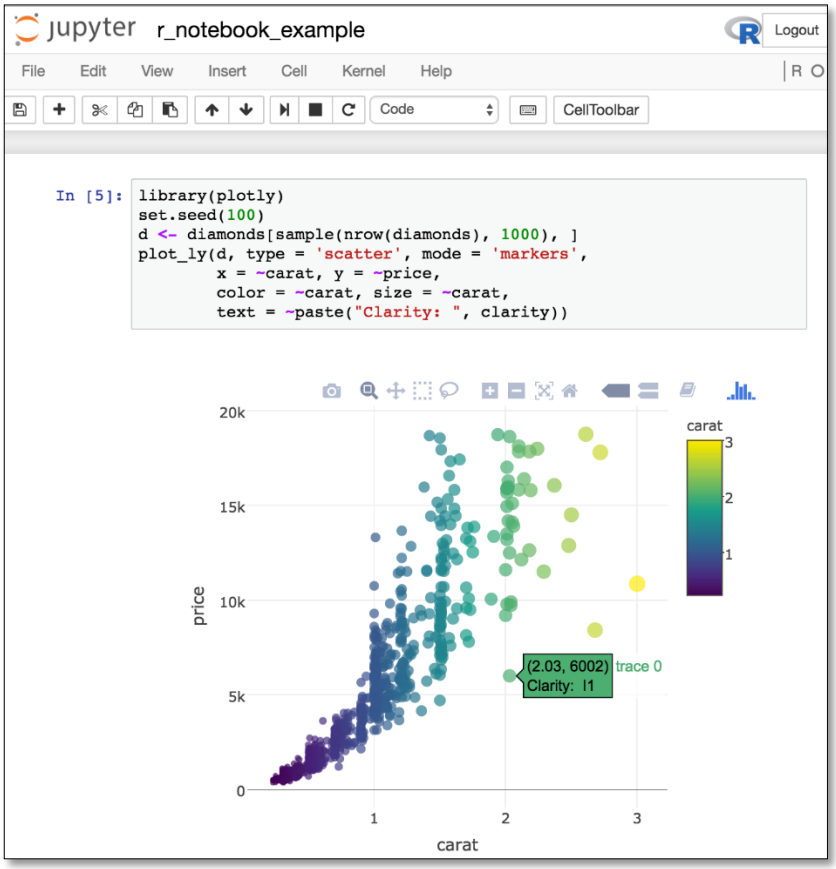


Рисунок 5 – Окно Jupyter Notebook

2.3 Используемые библиотеки

Помимо использования основных функций, реализованных в Python, также необходимо подключение ряда библиотек, представляющих собой набор модулей и функций, которые облегчают большое количество специфических операции с использованием этого языка программирования.

К таким библиотекам относятся:

1) Scikit-learn;

Одна из популярнейших библиотек для задач МО. Поддерживает многие алгоритмы МО такие как KNN, SVM и другие [29].

2) Pandas;

Библиотека, трансформирующая высокоуровневые структуры данных (например, датасеты) в простые в использовании и интуитивно понятные.

В ней находятся встроенные методы для группировки, комбинирования данных и их фильтрации, а также анализа временных рядов [14].

Важнейшая задача данной библиотеки в рамках МО – это получение данных в удобной пользователю форме из разнообразных источников (файлы с разрешение CSV, Excel, файлы БД SQL и т.д)

3) Seaborn;

Данная библиотека предназначена для создания статистической графики на Python. Он построен поверх matplotlib и тесно интегрирован со структурами данных Pandas [12].

4) NumPy;

NumPy предоставляет общие математические и числовые операции, в том числе методы для взаимодействия с матрицами и массивами [11].

5) XGBoost.

Специализированная библиотека, предназначенная в первую очередь для реализации метода “XGBoost” [13].

6) Snap7;

Библиотека, позволяющая осуществить соединение с ПЛК Simatic S7, в том числе операции по записи и чтению значений тегов, хранящихся в памяти ПЛК [45].

3 Подготовка исходных данных

Исходными данными для обучения являются параметры дефектов и заготовок, измеренные благодаря системе, описанной в разделе 1.2.

Данные представляют собой таблицу, где столбцы – это названия дефектов или их классов (типов), строки – значения параметров для одного измерения (одна заготовка).

Технологические теги параметров дефектов сведены в таблицу 1.

Таблица 1 – Технологические теги параметров

Название параметров				
X_Minimum	X_Perimeter	Length_of_Conveyer	Empty_Index	Outside_Global_Index
X_Maximum	Y_Perimeter	TypeOfSteel_A300	Square_Index	LogOfAreas
Y_Minimum	Sum_of_Luminosity	TypeOfSteel_A400	Outside_X_Index	Log_X_Index
Y_Maximum	Minimum_of_Luminosity	Steel_Plate_Thickness	Edges_X_Index	Log_Y_Index
Pixels_Areas	Maximum_of_Luminosity	Edges_Index	Edges_Y_Index	Orientation_Index
Luminosity_Index	SigmoidOfAreas	—	—	—

Таким образом, имеется 27 признаков, являющихся входной информацией для обучения и тестирования моделей. Формально признаком называется отображение $f : X \to D_f$, где D_f – множество допустимых значений признака [26].

В зависимости от природы множества *Df* признаки делятся на несколько типов:

А) $D_f = \{0, 1\}$, то f – бинарный признак;

Б) D_f – конечное множество, то f – номинальный признак;

В) D_f – конечное упорядоченное множество, то f – порядковый признак;

Г) D_f = R, то f — количественный признак.

От типа признака напрямую зависит выбор методов обработки того или иного столбца данных.

Оставшиеся 7 столбцов отображают принадлежность дефекта к определенному классу. Классы, представленные в выборке:

1) Pastry;

2) Z_Scratch;

3) K_Scatch;

4) Stains;

5) Dirtiness;

6) Bumps;

7) Other_Faults (прочие нарушения, не относятся к дефектам).

Результаты отнесения каждой из заготовок к определенному классу (данный процесс принято называть разметкой данных) являются выходным набором данных для обучаемых и тестируемых моделей.

Для осуществления операций по чтению и обработки данных, необходимо подключение библиотек (см. 2.3).

```
# Импортирование библиотек
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Рисунок 6 – Импортирование библиотек в программу

Работа с набором данных начинается с чтения файла, где они хранятся. Он представляет собой файл формата “csv”, где значения соседних признаков отделяются друг от друга запятой.

```
data = pd.read_csv('C:/Users/malya/Desktop/ВКР/Диплом/Data/defects.csv') # чтение файла
data.head() # обзор первых 5 строк файла
```

	X_Minimum	X_Maximum	Y_Minimum	Y_Maximum	Pixels_Areas	X_Perimeter	Y_Perimeter	Sum_of_Luminosity	Minimum_of_Luminosity	Maximum_of_Lumi
0	42	50	270900	270944	267	17	44.0	24220	76	
1	645	651	2538079	2538108	108	10	30.0	11397	84	
2	829	835	1553913	1553931	71	8	19.0	7972	99	
3	853	860	369370	369415	176	13	45.0	18996	99	
4	1289	1306	498078	498335	2409	60	260.0	246930	37	

5 rows × 34 columns

Рисунок 7 – Чтение файла с набором данных

Рассмотрим характеристики данных по каждому из признаков.

```
data.describe()
```

	X_Minimum	X_Maximum	Y_Minimum	Y_Maximum	Pixels_Areas	Y_Perimeter	Sum_of_Luminosity	Maximum_of_Luminosity	Length_of_Conveyer
count	1945.000000	1945.000000	1.945000e+03	1.945000e+03	1945.000000	1944.000000	1.945000e+03	1945.000000	1945.000000
mean	570.444216	617.202571	1.649672e+06	1.649726e+06	1890.681748	82.880144	2.059616e+05	130.204627	1459.608740
std	520.447843	497.473317	1.774542e+06	1.774554e+06	5163.631152	426.159416	5.118255e+05	18.675215	144.768744
min	0.000000	4.000000	6.712000e+03	6.724000e+03	2.000000	1.000000	2.500000e+02	37.000000	1227.000000
25%	51.000000	192.000000	4.683170e+05	4.685200e+05	84.000000	13.000000	9.522000e+03	124.000000	1358.000000
50%	434.000000	466.000000	1.199744e+06	1.199753e+06	175.000000	25.000000	1.922600e+04	127.000000	1364.000000
75%	1051.000000	1071.000000	2.183073e+06	2.183084e+06	812.000000	83.000000	8.256500e+04	140.000000	1650.000000
max	1705.000000	1713.000000	1.298766e+07	1.298769e+07	152655.000000	18152.000000	1.159141e+07	253.000000	1794.000000

8 rows × 32 columns

```
data.describe()
```

eel_A300	...	Orientation_Index	Luminosity_Index	SigmoidOfAreas	Pastry	Z_Scratch	K_Scatch	Stains	Dirtiness	Bumps	Other_Faults
5.000000	...	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000
0.400000	...	0.084002	-0.131297	0.585490	0.082776	0.097686	0.201028	0.037018	0.028278	0.207198	0.346015
0.490024	...	0.501154	0.148656	0.339325	0.275615	0.296966	0.400872	0.188854	0.165808	0.405403	0.475821
0.000000	...	-0.991000	-0.998900	0.119000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	...	-0.333300	-0.195000	0.248200	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	...	0.095200	-0.132800	0.506300	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1.000000	...	0.514300	-0.066600	0.999800	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
1.000000	...	0.991700	0.642100	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Рисунок 8 – Основные параметры значений признаков

Анализируя данные, можно прийти к выводу, что они однородны, т.е относятся к одному типу – количественный признак (не относится к меткам классов). Кроме того, для нескольких признаков разница значений в паре минимум – медиана и медиана – максимум отличаются в несколько раз (например, признак “Pixel_areas”, где максимальное значение признака 152655 при медиане 175). Это может свидетельствовать о наличие выбросов (аномальных, некорректных значениях) среди данных.

Перейдем к подготовке данных. Прежде чем приступить к обучению и тестированию модели необходимо убедиться, что данные обладают полнотой, т.е. не содержат некорректных значений и т.д.

1. Проверка полноты данных;

Таблица исходных данных не должна содержать пустых значений. В случае, если в строке в любом из признаков имеется хотя бы одно пустое значений, данная строка подлежит удалению, поскольку это может негативно отразиться на результатах обучения и тестирования, либо модель выдаст ошибку при обработке данных.

```
data.shape # Размеры исходного набора данных
(1945, 34)

data_mod_1 = data.dropna()  # Удаление строк с пустыми значениями и создание нового набора данных

data_mod_1.shape # Размеры нового набора данных
(1943, 34)
```

Рисунок 9 – Удаление строк (объектов) с пустыми значениями

Размер (длинна) набора данных после удаления строк с пустыми значениям уменьшился, проверка полноты данных выполнена.

2. Проверка некорректности значений;

Заключается в поиске аномальных значений – “выбросов”, которые также ухудшают качество обучения модели. К таковым относятся значения с чрезмерно большой или малой величиной, отрицательные значения (для параметров длина, площадь и т.д.), нулевые значения и т.д.

Выбросы могут искажать и сокращать информацию, содержащуюся в источнике данных или процедуре их генерации. В производстве наличие выбросов снижает результативность производственных процессов, качество продукции, а также процедур контроля продукции [29].

Для наглядности поиска и нахождения “выбросов” также может быть использован графический анализ, в частности построение диаграммы рассеивания. Демонстрация данной диаграммы на примере признака “Pixels_Areas” представлена на рисунке 10.

```
plt.scatter(data.index, data['Pixels_Areas']) # Построение диаграммы рассеивания для признака "Pixels_Areas"
```

```
<matplotlib.collections.PathCollection at 0x2d0f8409448>
```

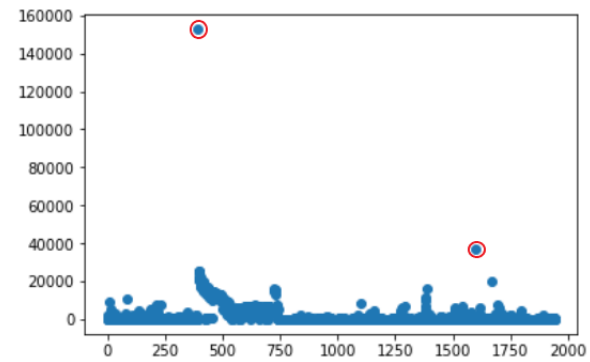


Рисунок 10 – Точечная диаграмма значений признака “ Pixels_Areas” (ось x – номер измерения, ось y – значение измерения)

Аналогично данный метод можно применить к остальным признакам, после чего удалять строки, содержащие выбросы, из набора данных.

На данный момент в теории статистического анализа нет однозначного критерия идентификации выбросов, поэтому в качестве аномальных значений могут выступать величины, выходящие за пределы ±3σ в случае нормального распределения данных,

Наиболее быстрый и простой метод очистки набора данных от выбросов заключается в нахождении значений процентилей “5” и “95” для всех признаков кроме тех, что обозначают классы дефектов, поскольку они принимают значения “0” или “1”. Затем происходит удаление строк со значениями признаков, меньшими или большими соответствующих процентилей.

```
# Выделение набора данных с признаками, имеющими явные выбросы
data_mod_2 = data_mod_1.drop(['Pastry','Z_Scratch','K_Scatch','Stains','Dirtiness','Bumps','Other_Faults',
                              'TypeOfSteel_A300','TypeOfSteel_A400','Outside_Global_Index',
                              'Steel_Plate_Thickness','Edges_Y_Index','SigmoidOfAreas'], axis='columns')
# Выделение набора данных с признаками без явных выбросов
norm_features = pd.DataFrame
norm_features = data_mod_1.drop(data_mod_2.columns, axis='columns')
#norm_features.info()
# Обозначение процентилей
low = .05
high = .95
# Формирование набора данных с посчитанными для каждого признака процентилями
quant_df = data_mod_2.quantile([low, high])
# Формирование нового набора данных со значениями признаков, входящих в доверительный интервал
data_mod_2 = data_mod_2.apply(lambda x: x[(x>quant_df.loc[low,x.name]) & (x < quant_df.loc[high,x.name])], axis=0)
# Объединение наборов данных data_mod_2 и norm_features
data_mod_3 = pd.concat([data_mod_2,norm_features],axis=1)
data_mod_3.info()
```

Рисунок 11 – Очистка набора данных от выбросов

```
data_mod_4 = data_mod_3.dropna()  # Удаление строк с пустыми значениями и создание итогового набора данных
data_mod_4.shape
```

```
(1422, 34)
```

```
data_mod_4.to_csv('defects_filt.csv') # Сохранение отфильтрованного набора данных в виде *csv файла
```

Рисунок 12 – Сохранение набора данных без выбросов

Размеры полученного набора данных уменьшились, что говорит об успешности проведенной фильтрации от выбросов.

3. Проверка важности признаков;

Определение степени влияния каждого из признаков может быть довольно полезно, поскольку, во-первых, “слабые” информационные признаки могут непродуктивно нагружать вычислительные ресурсы, не привнося полезной информации (особенно полезно в случае, когда признаков несколько сотен и более). Во-вторых, нахождение сильной корреляции между определенными признаками и классами дефектов дает возможности экспертам сделать предположения относительно возможных причин возникновения дефекта.

Оценка важности параметров признаков может быть проведена при помощи инструмента “HeatMap”.

```
# Построение корреляционной матрицы
plt.figure(figsize=(20, 20))
sns.heatmap(data_mod_4.corr(), annot = True, vmin=-1, vmax=1, center= 0, fmt='.1g', cmap= 'coolwarm')
```

Рисунок 13 – Построение “HeatMap”

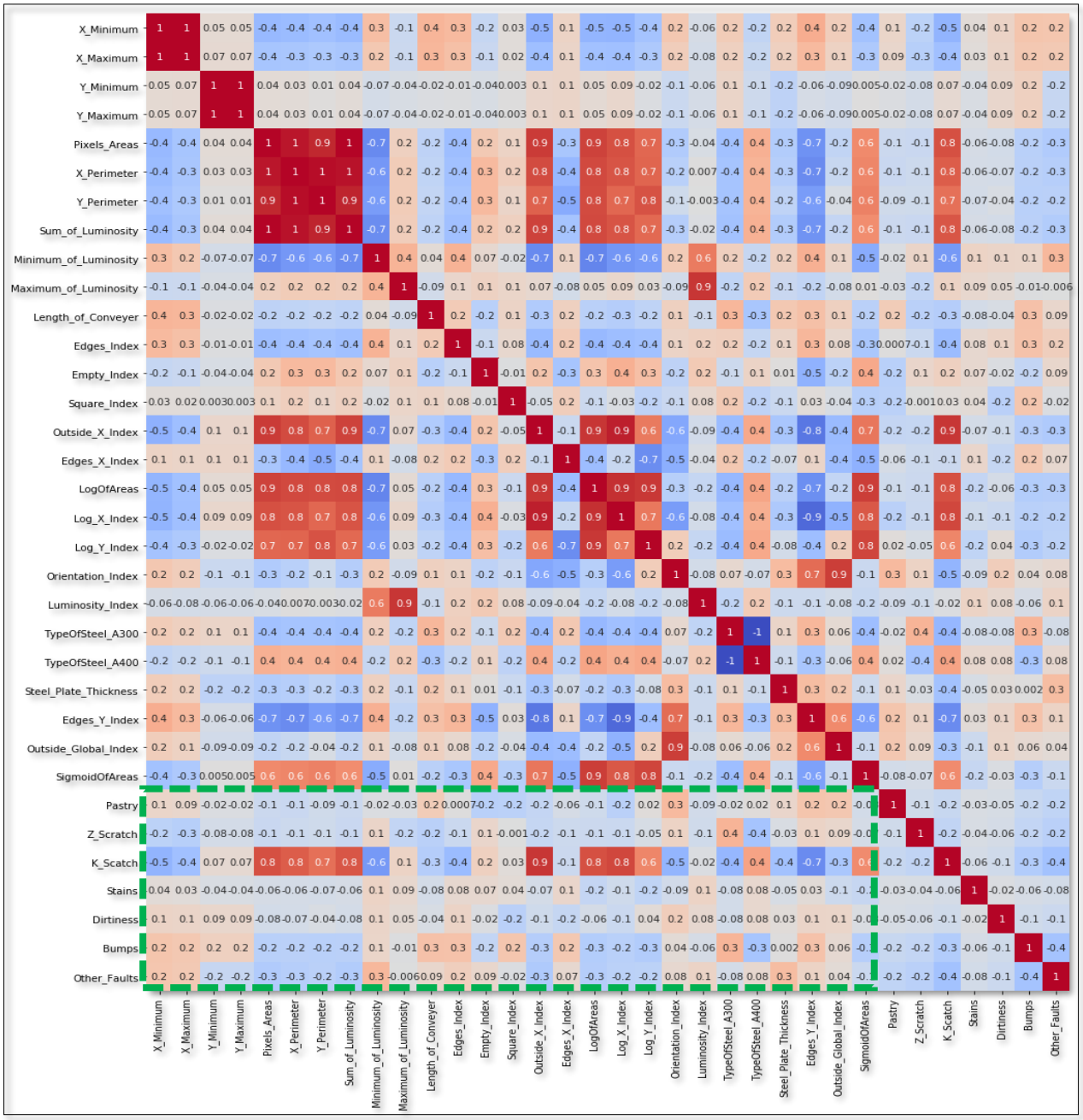


Рисунок 14 – Матрица корреляций

Визуальный анализ матрицы корреляций показывает, что каждый из признаков имеет как минимум для одного из классов дефектов коэффициент корреляции $|k| \geq 0.2$, что означает наличие уже минимальной связи между параметрами.

Необходимо отметить, что на основании этих данных можно утверждать лишь о степени связи между переменными, но не о существовании причинно-следственной зависимости между ними [17].

4. Нормализация или стандартизация данных;

В зависимости от применяемого метода зачастую требуется провести нормализацию данных, поскольку большинство градиентных методов величинам измерений признака, поэтому в качестве дополнительной обработки данных используется нормализация или стандартизация [9].

Нормализация предполагает замену значений признаков так, чтобы каждый из них лежал в диапазоне от 0 до 1. Каждое значение в данном случае находится следующим образом:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}, \quad (1)$$

где *x* – исходное значение признака, *min(x)* – минимальное значение признака, *max(x)* – максимальное значение признака.

Стандартизация же подразумевает такую предобработку данных, после которой каждый признак имеет среднее 0 и дисперсию 1.

$$z = \frac{x - \mu}{\sigma}, \quad (2)$$

где μ – матожидание значений признака, σ – стандартное отклонение.

Реализация приведенных методов обработки данных непосредственно в Python будет показана в разделе 2.3.3.

5. Кодирование данных;

Большинство методов машинного обучения ожидают, что данные, поступающие на вход, будут параметрами с числовым форматом, а не текстовым.

Кроме того, в дальнейшем для обучения и тестирования моделей будет необходим столбец, содержащий в себе информацию о всех классах дефектов.

Следовательно, необходимо преобразовать такие метки в числовые метки. Этот процесс называется кодированием меток.

Определение формата имеющихся данных представлено на рисунке 15.

```
data_mod_4.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1422 entries, 0 to 1940
Data columns (total 34 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   X_Minimum               1422 non-null   float64
 1   X_Maximum               1422 non-null   float64
 2   Y_Minimum               1422 non-null   float64
 3   Y_Maximum               1422 non-null   float64
 4   Pixels_Areas            1422 non-null   float64
 5   X_Perimeter             1422 non-null   float64
 6   Y_Perimeter             1422 non-null   float64
 7   Sum_of_Luminosity       1422 non-null   float64
 8   Minimum_of_Luminosity   1422 non-null   float64
 9   Maximum_of_Luminosity   1422 non-null   float64
 10  Length_of_Conveyer      1422 non-null   float64
 11  Edges_Index             1422 non-null   float64
 12  Empty_Index             1422 non-null   float64
 13  Square_Index            1422 non-null   float64
 14  Outside_X_Index         1422 non-null   float64
 15  Edges_X_Index           1422 non-null   float64
 16  LogOfAreas              1422 non-null   float64
 17  Log_X_Index             1422 non-null   float64
 18  Log_Y_Index             1422 non-null   float64
 19  Orientation_Index       1422 non-null   float64
 20  Luminosity_Index        1422 non-null   float64
 21  TypeOfSteel_A300        1422 non-null   int64
 22  TypeOfSteel_A400        1422 non-null   int64
 23  Steel_Plate_Thickness   1422 non-null   int64
 24  Edges_Y_Index           1422 non-null   float64
 25  Outside_Global_Index    1422 non-null   float64
 26  SigmoidOfAreas          1422 non-null   float64
 27  Pastry                  1422 non-null   int64
 28  Z_Scratch               1422 non-null   int64
 29  K_Scatch                1422 non-null   int64
 30  Stains                  1422 non-null   int64
 31  Dirtiness               1422 non-null   int64
 32  Bumps                   1422 non-null   int64
 33  Other_Faults            1422 non-null   int64
```

Рисунок 15 – Типы данных признаков и классов

Данные имеют форматы: “*int64*” – целочисленный тип или “*float64*” – вещественные числа двойной точности, где 1 бит знака, 11 бит экспоненты, 52 бита мантисы (числа размером 8 байт).

Поскольку все данные являются числами, то кодирование данных не требуется.

6. Формирование набора меток.

Для использования набора данных для дальнейшего обучения моделей и их тестирование необходимо сформировать единый столбец меток, т.е столбец, содержащий информацию обо всех имеющихся классах дефектов.

```
y = [] # Сформируем список , куда будут записываться дефекты, если они есть
for i in range(data_mod_4.shape[0]): # Проверка наличия дефекта в каждой строке, при наличии код дефекта добавляется в список
    if data_mod_4["Pastry"].values[i] == 1:
        y.append(1)
    elif data_mod_4["Z_Scratch"].values[i] == 1:
        y.append(2)
    elif data_mod_4["K_Scatch"].values[i] == 1:
        y.append(3)
    elif data_mod_4["Stains"].values[i] == 1:
        y.append(4)
    elif data_mod_4["Dirtiness"].values[i] == 1:
        y.append(5)
    elif data_mod_4["Bumps"].values[i] == 1:
        y.append(6)
    else:
        y.append(7)
y = np.array(y) # Создание массива на базе списка, который будет являться основой для формирования набора данных
defects_type = pd.DataFrame({'defect':y}) # Создание набора данных со столбцом "defect", где содержаться названия дефектов
data_mod_4.reset_index(inplace = True) # Упорядочить заново индексы в связи с удалением строк на пред-их этапах
data_mod_5 = data_mod_4.drop(data_mod_4.columns[0], axis='columns')
data_mod_6 = data_mod_5.join(defects_type) # Объединение матрицы признаков и меток класса в 1 набор данных
data_final = data_mod_6.drop(columns = defects) # Удаление столбцов с отдельными дефектами
data_final.to_csv('data_final.csv') # Сохранение итогового набора данных в файл формата .csv
```

```
data_final.head()
```

Orientation_Index	Luminosity_Index	TypeOfSteel_A300	TypeOfSteel_A400	Steel_Plate_Thickness	Edges_Y_Index	Outside_Global_Index	SigmoidOfAreas	defect
0.8182	-0.2913	1	0	80	1.0000	1.0	0.5822	1
0.7931	-0.1756	1	0	80	0.9667	1.0	0.2984	1
0.8444	-0.1568	0	1	290	1.0000	1.0	0.5212	1
0.8736	-0.2267	0	1	40	1.0000	1.0	0.9874	1
0.5000	0.1841	0	1	150	1.0000	1.0	0.3359	1

Рисунок 16 – Создание конечного набора данных

В результате получен итоговый набор данных, содержащий в явном виде набор признаков и меток. Расшифровка значений меток классов представлена в таблице 2.

Таблица 2 – Соответствие меток и классов дефектов

Значение в столбце “defect”	Класс дефекта
1	Pastry
2	Z_Scratch
3	K_Scratch
4	Stains
5	Dirtiness
6	Bumps
7	Other_faults

Использование методов обработки данных, представленных в текущем разделе, позволяют использовать набор данных для дальнейшего создания моделей, их обучения и тестирования.

Исходя из основных целей выполнения работы, а именно детектирование и классификация дефектов, необходимо выполнить

последнюю операцию, заключающуюся в следующем: поскольку для детектирования необходимо только установить наличие или отсутствие дефекта, то классы с кодами 1-6 образуют новый класс – “Наличие дефекта” (код 0) и “Отсутствие дефекта” (код 1).

Для классификации дефекты данные по классу “Other faults” необходимо удалить, поскольку они не относятся к конкретному типу дефекта.

Рисунок 17 – Диаграмма представителей объектов классов 1 – 6 в конечном наборе данных (ось *x* – метки классов, ось *y* – количество объектов каждого класса)

```
# Отрисовка гистограммы классов в наборе данных для классификации
sns.set(rc={'axes.labelsize':30, 'xtick.labelsize':25,'ytick.labelsize':20}) # Задание размеров подписей на гистограмме
plt.figure(figsize=(12, 6)) # Размеры фигуры
sns.set_style('whitegrid') # Установка стиля гистограммы
sns.countplot(data['defect']) # Выбор данных на гистограмме
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1988af1e988>
```

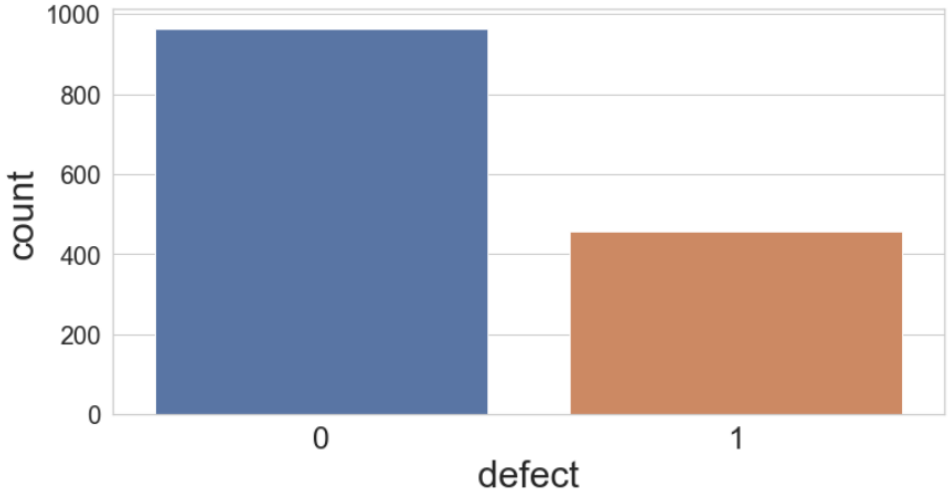


Рисунок 18 – Диаграмма представителей объектов классов 0 – 1 в конечном наборе данных (ось *x* – метки классов, ось *y* – количество объектов каждого класса)

Анализ количества объектов различных классов показывает, что классы не сбалансированы. Данное свойство набора данных влияет на выбор метрики качества при оценке результатов и на отношение к предсказаниям относительно малочисленных классов в целом.

4 Реализация методов машинного обучения

Практическая часть представляет собой описание задачи классификации, особенности и методов ее решения. Далее будет представлен процесс подготовки тренировочного и тестового наборов данных, на основе которых будут обучаться различные модели и оцениваться их точность.

4.1 Описание задачи классификации

Машинное обучение – это обучение компьютерной программы или алгоритма постепенному улучшению исполнения поставленной задачи.

Машинное обучение состоит из множества математических, статистических и вычислительных методов для разработки алгоритмов, способных решить задачу не прямым способом, а на основе поиска закономерностей в разнообразных входных данных.

Решение вычисляется не по четкой формуле, а по установленной зависимости результатов от конкретного набора признаков и их значений [19].

Классификация – это один из разделов машинного обучения. В общих чертах он описывается следующим образом: имеется обучающая выборка, в которой представлены объекты в виде их признакового описания (вектор признаков) и метки класса. Задача классификации заключается в нахождении алгоритма, который для каждого нового объекта (его признакового описания) определит метку класса этого объекта. Это эквивалентно построению разделяющей поверхности в многомерном признаковом пространстве [20].

Более формально задача классификации может быть описана так: пусть заданы K классов: $C_0, C_1, ..., C_{k-1}$. Пусть U – множество объектов, каждый из которых принадлежит одному из классов $C_0, C_1, ..., C_{k-1}$. Функция μ: $U \rightarrow R^D$ вычисляет на основе объекта u вектор характеристик размерности D.

Определение: задача определения, какому из K классов $C_0, C_1, ..., C_{k-1}$ принадлежит объект $u \in U$ на основе его характеристик, представленных D-мерным вектором x входных переменных называется задачей классификации.

Пусть обучающая выборка построена на основе данных N объектов u_1, ..., $u_N \in U$ и состоит соответственно из N наблюдений $\{x_n\}$, где $x_n = \mu(u_n)$, $n = 1, ..., N$. Для каждого входного вектора $x_n = \mu(u_n)$ в обучающей выборке хранится номер $t_n \in \{0, ..., K - 1\}$ класса, которому принадлежит объект u_n. Задача – по вектору характеристик x объекта u определить класс номер класса t, которому принадлежит объект u [21].

Решение задачи классификации часто можно разделить на два этапа:

1) Вывод: вычисление условных вероятностей $p\ (u \in C_i \mid \mu(u) = x)$, $i = 0, ..., K - 1$;

2) Принятие решения (англ. decision stage): на основе условных вероятностей $p\ (u \in C_i \mid \mu(u) = x)$, $i = 0, ..., K - 1$, принимается решение о принадлежности объекта u к тому или иному классу:

$$t = arg\ max_i\ p\ (u \in C_i \mid \mu(u) = x) \quad (3)$$

4.2 Метрики качества в задачах классификации

Прежде чем перейти непосредственно к метрикам необходимо ввести концепцию для описания этих метрик в терминах ошибок классификации – confusion matrix (матрица ошибок).

Допустим, что имеется два класса и алгоритм, предсказывающий принадлежность каждого объекта одному из классов, тогда матрица ошибок классификации будет выглядеть следующим образом.

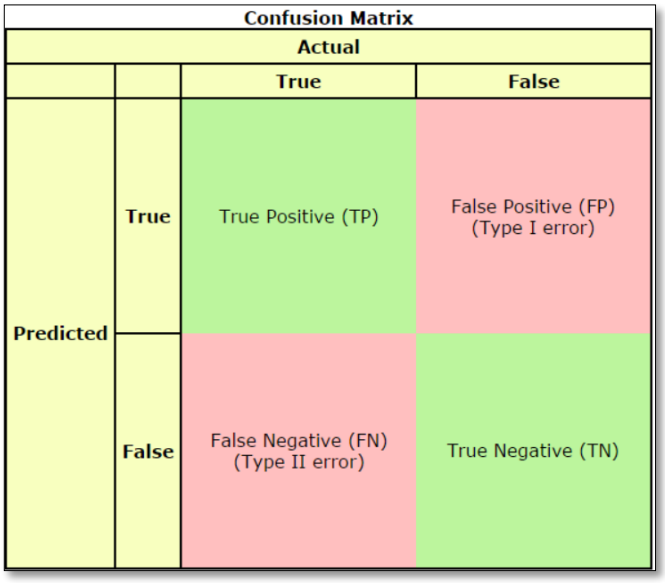


Рисунок 19 – Матрица ошибок

Здесь “Predicted” – это предсказание алгоритма по i-му объекту, “Actual” – истинная метка класса на i-ом объекте. Таким образом, ошибки классификации бывают двух видов: False Negative (FN) и False Positive (FP).

Тогда справедливо следующее выражение:

$$True\ Positives + True\ Negatives + False\ Positives + False\ Negatives = N, \quad (4)$$

где N – количество объектов в наборе данных.

Количественные показатели, связанные с ошибками классификации:

1) Ошибки первого и второго рода:

– Ошибка первого рода, ложная положительная классификация, "ложная тревога":

$$\alpha = False\ Positives\ /\ (True\ Negatives + False\ Positives) \quad (5)$$

– Ошибка второго рода, ложная отрицательная классификация, "пропуск цели":

$$\beta = False\ Negatives\ /\ (True\ Positives + False\ Negatives) \quad (6)$$

2) Accuracy (достоверность):

$$Accuracy = (True\ Positives + True\ Negatives)\ /\ N \quad (7)$$

“Accuracy” является наиболее интуитивно понятной метрикой, отражает долю правильных ответов алгоритма.

3) Precision and recall (точность и полнота):

$$Precision = True\ Positives / (True\ Positives + False\ Positives) \quad (8)$$

“Precision” является долей объектов, названных классификатором положительными и при этом действительно являющимися таковыми.

$$Recall = True\ Positives / (True\ Positives + False\ Negatives) \quad (9)$$

“Recall” показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм.

4) F1-score:

$$F1\ Score = 2*(Precision \cdot Recall) / (Precision + Recall) \quad (10)$$

где b – в данном случае определяет вес точности в метрике, и при b =1 это среднее гармоническое (с множителем 2, чтобы в случае “precision” = 1 и “recall” = 1 иметь $F1$ = 1).

F-мера достигает максимума при полноте и точности, равными единице, и близка к нулю, если один из аргументов близок к нулю.

Таким образом $F1$ метрика агрегирует “precision” и “recall” в единый критерий качества, F-мера – среднее гармоническое “precision” и “recall”.

4.3 Формирование тренировочного и тестового наборов данных

Перед тем как создать тренировочный и тестовые наборы данных необходимо сформировать матрицы входных признаков и матрицу выходов.

```
X = data.drop('defect', axis = 1) # Формирование входного набора данных
Y = data['defect'] # Формирование выходного набора данных
X.shape, Y.shape # Размерности матриц X и Y
```

```
((1422, 27), (1422,))
```

Рисунок 20 – Формирование набора признаков X и столбца Y

Разбиение матриц на тренировочную и тестовую часть осуществляется при помощи функции “train_test_split”, которая в заданном пользователем соотношении в случайном порядке относит строки к одной либо другой части.

```
from sklearn.model_selection import train_test_split # Импорт из библиотеки sklearn инструмента train_test_split
# Формирование тренировочных и тестовых выборок для X и Y
x_train, x_test, y_train, y_test = train_test_split (X, Y, test_size = 0.3, random_state = 17)
x_train.shape, x_test.shape, y_train.shape, y_test.shape # Размерности выборок
```

```
((995, 27), (427, 27), (995,), (427,))
```

Рисунок 21 – Разбиение данных на тренировочную и тестовую часть

Таким образом имеется матрица входных признаков размерностью 995 x 27 и матрицу меток класса размерностью 427 x 1.

Поскольку часть методов используют меры расстояния, то все значения признаков должны находиться в одном масштабе. Для этого используется метод стандартизации, описанный в главе 3.

```
from sklearn.preprocessing import StandardScaler # Импорт функции стандартизации данных
Scaler = StandardScaler() # Объявление функции
Scaler.fit(x_train.values.reshape(-27, 27)) # Подача на вход стандартизатора тренировочных данных
Scaler.fit(x_test.values.reshape(-27, 27)) # Подача на вход стандартизатора тестовых данных
x_train_scaled = Scaler.transform(x_train.values.reshape(-27, 27)) # Преобразование тренировочных данных
x_test_scaled = Scaler.transform(x_test.values.reshape(-27, 27)) # Преобразование тестовых данных
```

```
x_train_scaled, x_test_scaled
```

```
(array([[ 0.67160118,  0.63758457, -0.39250406, ...,  0.25174936,
         -1.23556757, -1.34115391],
        [-0.99008578, -0.82068816, -0.58216068, ..., -1.41616408,
         -1.23556757,  1.17987068],
        [-0.7900679 , -0.89138005, -0.98252236, ..., -0.0044041 ,
         -1.23556757, -0.98026856],
        ...,
        [-0.73429369, -0.82068816, -0.54482942, ...,  0.49220165,
          0.80934465,  1.15461479],
        [ 1.55052588,  1.57071753, -0.53316922, ..., -0.67731162,
         -1.23556757, -0.8043902 ],
        [ 1.67553705,  1.72623969,  2.63800448, ..., -0.83252896,
         -1.23556757, -0.7295354 ]]),
 array([[-0.99393227, -0.78635209,  0.43084945, ..., -1.60278376,
         -1.23556757,  1.17987068],
        [-0.96316029, -1.08123827,  0.02387619, ...,  0.54782867,
          0.80934465, -1.16679699],
        [ 1.13318089,  1.12434873,  0.58604144, ...,  0.89280592,
          0.80934465, -0.65285486],
        ...,
        [-0.20540027, -0.16022392,  0.01551303, ..., -1.57452165,
         -1.23556757,  1.17987068],
        [ 0.33310939,  0.28614488, -0.52640585, ...,  0.65683969,
          0.80934465, -0.68176221],
        [-1.04201349, -1.16404877, -0.95319445, ...,  0.89280592,
          0.80934465, -0.74840124]]))
```

Рисунок 22 – Стандартизация данных

4.4 Используемые методы классификации

В контексте машинного обучения классификация относится к обучению с учителем. Такой тип обучения подразумевает, что данные, подаваемые на входы системы, уже помечены, а важная часть признаков уже разделена на отдельные категории или классы [24].

Цель алгоритма обучения с учителем — на основе набора данных создать модель, которая принимает вектор признаков x на входе и возвращает информацию, которая позволяет определить метку для этого вектора признаков.

В данной работе будут рассмотрены и применены следующие методы классификации:

1) метод k-ближайших соседей (K-Nearest Neighbors или KNN);
2) метод опорных векторов (Support Vector Machines или SVM);
3) дерево решений (Decision Tree Classifier)
4) наивный байесовский метод (Naive Bayes);
5) логистическая регрессия (Logistic Regression);
6) экстремальный градиентный бустинг (XGBoost).

Методы 1 – 5 осуществляются благодаря использованию библиотеки Scikit-Learn, которой находится большое количество алгоритмов для задач, связанных с классификацией и машинным обучением в целом.

Для использования метода “XGBoost” необходим импорт одноименной библиотеки.

4.4.1 Дерево решений

Дерево принятия решений – средство поддержки принятия решений, которое использует древовидный граф.

В общем случае – это *k*-ичное дерево с решающими правилами в нелистовых вершинах (узлах) и некотором заключении о целевой функции в листовых вершинах (прогнозом). Решающее правило – некоторая функция от объекта, позволяющее определить, в какую из дочерних вершин нужно поместить рассматриваемый объект [22].

Основным алгоритмом разбиения в дереве решений является принцип максимизации прироста информации – на каждом шаге выбирается тот признак, при разделении по которому прирост информации оказывается наибольшим. Дальше процедура повторяется рекурсивно, пока энтропия не окажется равной нулю или какой-то малой величине.

Формально решение задачи деревом решений выглядит следующим образом. Пусть имеется K классов: C_1, C_2, ..., C_k. Через $x \in R^D$ обозначим вектор входных переменных X. Пусть обучающая выборка содержит N элементов,

через S_i обозначим множество элементов обучающей выборки, достигших i-го узла при обучении, через S_{ij} – множество элементов обучающей выборки, достигших j-го узла, являющегося потомком i-го узла. Через $S(k)$ обозначим множество элементов обучающей выборки, соответствующих объектам, принадлежащим классу C_k.

Функция выбора характеристик (разделяющая функция) – это функция, получающая на вход вектор входных переменных X и возвращающая вектор, содержащий подмножество входных переменных вектора X.

Для определения того, какую разделяющую функцию выбрать для данного узла, необходимо оценить качество разделения выборки для каждой из функции посредством критерия прироста информации:

$$I = H(S_i) - \sum_j \frac{N_{ij}}{N_i} * H\left(S_{ij}\right), \tag{11}$$

где энтропия $H(S_i)$:

$$H(S_i) = -\sum_k \frac{N_i^{(k)}}{N_i} * \log\left(\frac{N_i^{(k)}}{N_i}\right), \tag{12}$$

где $N_{ij} = \left|S_{ij}\right|,\ N_i = |S_i|, N_i^{(k)} = \left|S_i{}^{(k)}\right|$.

Энтропия соответствует степени хаоса в системе. Чем выше энтропия, тем менее упорядочена система и наоборот.

Алгоритм работает следующим образом. Пусть S обозначает множество размеченных данных. В начале дерево решений имеет только начальный узел, который содержит все данные: $S = \{(x_i, y_i)\}_{i=1}^N$.

Затем осуществляется перебор всех признаков $j = 1, ..., D$ и всех порогов t, далее множество S разбивается на два подмножества: $S_- = \{(x,y)|(x,y) \in S, x^{(j)} < t\}$ и $S_+ = \{(x,y)|(x,y) \in S, x^{(j)} \geq t\}$.

Два новых подмножества образуют два новых листовых узла, и осуществляется оценка всех возможных пар (j, t), насколько хорошим получилось расщепление на части S_- и S_+. Наконец, происходит выбор наилучших значений (j, t), разбиваем S на S_- и S_+, формируется два новых листовых узла и процесс деления повторяется.

На рисунке 23 представлено графическая структура дерева.

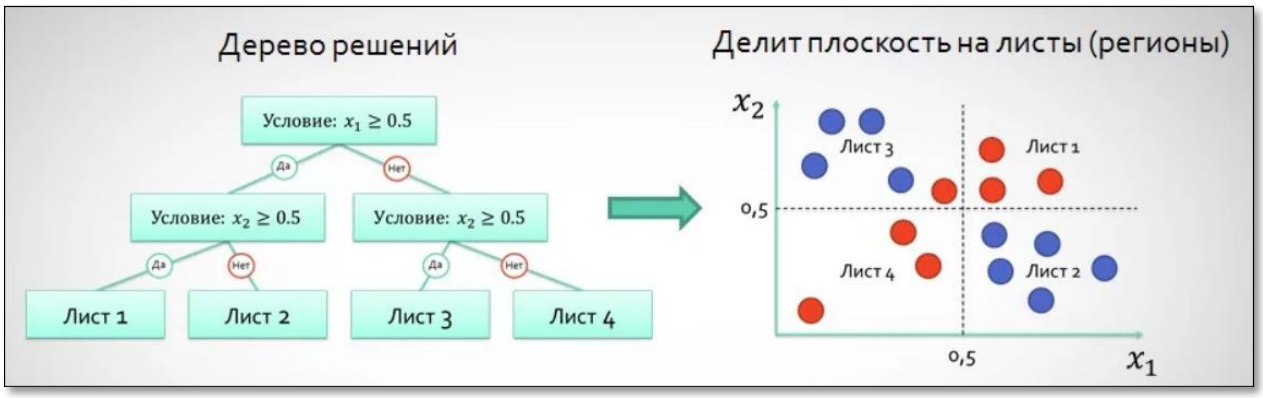


Рисунок 23 – Иллюстрация работы “Дерева решений”

К основным гиперпараметрам метода “Дерево решений” относятся:

1) max_depth – максимальная глубина дерева;

Точка, на которой останавливается разбиение узлов, аналогично выбору максимального количества слоев в глубокой нейронной сети. Меньшее количество сделает модель более быстрой, но не точной. Большее количество увеличивает точность, но создает риски переобучения и замедляет процесс.

2) max_features – число признаков.

Необходимо для поиска лучшей точки для разбиения. Чем больше количество признаков, тем точнее результат. Однако обучение занимает больше времени.

Параметры дерева (и большинства других методов) настраиваются в зависимости от входных данных.

Далее на вход дерева с наилучшими параметрами поступает тестовая выборка для определения результатов и качества предсказаний.

Одним из преимуществ данного метода является тот факт, что метод работоспособен как в случае бинарной, так и многоклассовой классификации.

Обучение и тестирование модели для идентификации представлено на рисунках 24, 25 и 26, а классификация на рисунках 28 и 29.

```
# Реализация дерева решений
from sklearn.tree import DecisionTreeClassifier # Импорт класса Дерево решений
from sklearn.model_selection import GridSearchCV # Импорт сети параметров для сравнения модели с разными признаками
TreeClassifier = DecisionTreeClassifier(random_state = 17) # Объявления классификатора
tree_params = {'max_depth': np.arange(1,10), 'max_features': [0.5,0.7,1]} # Установка диапазонов поиска параметров дерева
tree_grid = GridSearchCV(TreeClassifier, tree_params) # Заполнение сети деревьями со всеми вариантами параметров
tree_grid.fit(x_train, y_train) # Обучение всех деревьев в сети
```

```
GridSearchCV(cv=None, error_score=nan,
             estimator=DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
                                              criterion='gini', max_depth=None,
                                              max_features=None,
                                              max_leaf_nodes=None,
                                              min_impurity_decrease=0.0,
                                              min_impurity_split=None,
                                              min_samples_leaf=1,
                                              min_samples_split=2,
                                              min_weight_fraction_leaf=0.0,
                                              presort='deprecated',
                                              random_state=17,
                                              splitter='best'),
             iid='deprecated', n_jobs=None,
             param_grid={'max_depth': array([1, 2, 3, 4, 5, 6, 7, 8, 9]),
                         'max_features': [0.5, 0.7, 1]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0)
```

```
tree_grid.best_score_,  tree_grid.best_params_ # Выдача лучше результата обучения дерева и его параметров
```

```
(0.7668341708542714, {'max_depth': 9, 'max_features': 0.5})
```

Рисунок 24 – Обучение модели “Дерево решений” (детектирование)

```
# Тестирование модели и проверка качества предсказания
from sklearn.metrics import accuracy_score # Импорт метрики "Доля правильных ответов"
y_tree_pred = tree_grid.predict(x_test) # Получение предсказаний модели
accuracy_score(y_tree_pred, y_test) # Проверка качества
```

```
0.7517564402810304
```

```
print(y_tree_pred) # Вывод предсказанных классов для всех элементов тестовой выборки
print(y_tree_pred[0]) # Вывод предсказанного класса для конкретного элемента тестовой выборки
```

```
[0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 1 1 1 0 0 0
 1 1 1 1 0 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0
 0 0 1 0 1 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0
 0 0 0 0 1 0 0 1 1 1 1 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0
 0 1 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0
 0 0 0 0 0 0 1 0 0 1 0 1 0 1 1 1 0 1 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 0 1 0
 1 1 0 0 0 0 0 1 0 1 1 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 0 1 1 0 0
 1 1 1 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0
 1 0 0 1 0 0 0 0 0 0 0 0 1 1 0 1 0 1 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1
 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 1 0 0 1 1
 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0
 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1]
0
```

Рисунок 25 – Результаты тестирования модели “Дерево решений” (детектирование)

```
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test,y_tree_pred, target_names=['0','1']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           0       0.83      0.81      0.82       296
           1       0.59      0.62      0.60       131

    accuracy                           0.75       427
   macro avg       0.71      0.71      0.71       427
weighted avg       0.76      0.75      0.75       427
```

```
# Вызов инструмента для графической постройки дерева
from sklearn.tree import export_graphviz
export_graphviz(tree_grid.best_estimator_, out_file = 'C:/Users/malya/Desktop/ВКР/Диплом/default_tree_2.dot',
                feature_names = x_train.columns, filled = True) # Сохранение файла
```

Рисунок 26 – Метрики качества для модели “Дерево решений” (детектирование)

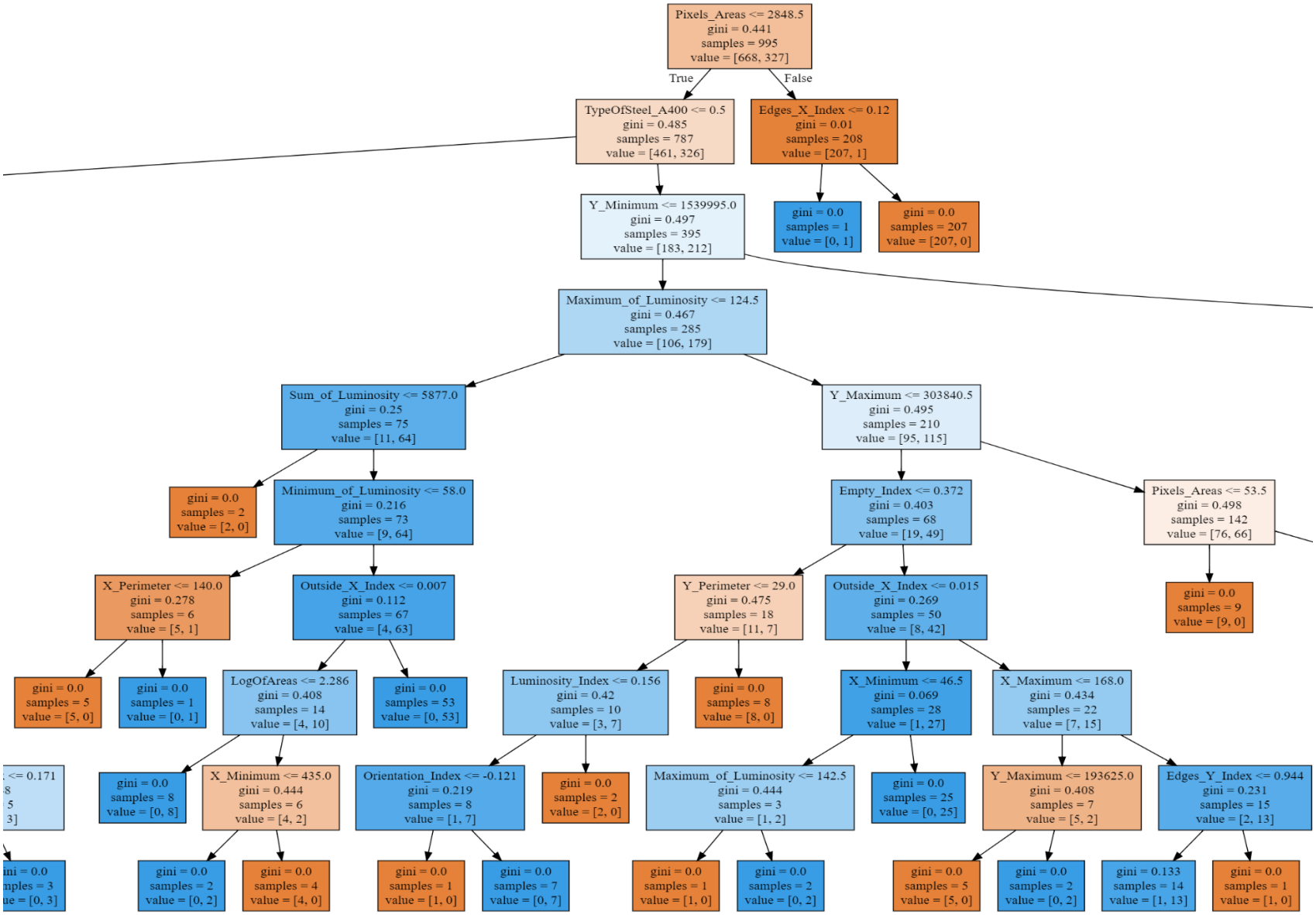


Рисунок 27 – Фрагмент дерева решений

```
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test,y_tree_pred, target_names=['1','2','3','4','5','6']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           1       0.50      0.58      0.54        24
           2       0.86      0.84      0.85        45
           3       0.94      0.97      0.96       100
           4       0.78      0.70      0.74        10
           5       0.81      0.93      0.87        14
           6       0.81      0.75      0.78        97

    accuracy                           0.83       290
   macro avg       0.78      0.80      0.79       290
weighted avg       0.84      0.83      0.84       290
```

Рисунок 28 – Метрики качества для модели “Дерево решений” (классификация)

```
print(y_tree_pred)
print(y_tree_pred[16])
```

```
[2 6 6 3 3 2 3 6 4 6 5 3 6 6 6 3 3 3 6 3 3 6 2 6 3 1 4 3 2 2 2 6 6 3 2 6 6
 3 6 1 4 3 3 6 3 3 6 3 3 1 6 1 1 3 6 3 6 3 4 1 6 1 6 1 6 3 6 3 1 6 3 6 2 3
 3 1 6 3 2 6 6 5 1 6 3 2 3 1 6 6 3 2 2 4 3 6 6 1 3 3 3 3 3 2 2 3 3 6 6 4 3
 4 6 3 2 6 6 1 6 2 3 3 3 5 3 6 3 3 6 3 3 5 3 6 2 2 2 6 6 3 6 6 6 3 5 6 2 3
 1 5 3 6 3 6 3 3 6 1 1 6 4 6 1 2 2 6 6 3 3 6 6 2 3 3 2 3 6 5 3 5 2 1 6 3 3
 3 2 5 3 6 2 6 3 1 6 3 6 6 2 3 3 1 1 2 5 3 6 3 6 5 6 6 3 6 5 3 6 3 6 1 2 6
 6 2 6 3 6 2 3 1 6 5 6 6 5 6 3 1 3 2 3 3 2 3 6 6 2 6 3 2 5 3 3 3 2 2 6 3 2
 3 5 6 1 2 6 6 3 2 3 3 3 2 6 3 1 6 6 3 3 2 3 3 3 2 3 3 3 3 1 4]
3
```

Рисунок 29 – Результаты тестирования модели “Дерево решений” (классификация)

4.4.2 Метод ближайших соседей

Метод ближайших соседей – метрический алгоритм, суть которого заключается в рассмотрении нескольких ближайших в некоторой метрике соседей с последующим выбором доминирующего среди них класса [19].

Иными словами, алгоритм находит ближайший классифицируемому объект из обучающей выборки, смотрит на его класс и относит классифицируемый объект к этому классу.

Формальное описание алгоритма выглядит следующим образом:

Пусть задана обучающая выборка: $X^m = \{(x_1, y_1), \dots, (x_m, y_m)\}$.

Пусть на множестве объектов задана функция расстояния $\rho(x, x')$. Чем больше значение этой функции, тем менее схожими являются два объекта (x, x').

Для произвольного объекта u расположим объекты обучающей выборки x_i в порядке возрастания расстояний до u:

$$\rho(u, x_1; u) \leq \rho(u, x_2; u) \leq \cdots \leq \rho(u, x_m; u), \quad (13)$$

где через x_i; u обозначается тот объект обучающей выборки, который является i-м соседом объекта u. Аналогичное обозначение вводится и для ответа на i-м соседе: y_i; u.

В наиболее общем виде алгоритм ближайших соседей есть:

$$a(u) = argmax \sum_{i=1}^{m} [x_{i;u} = y] w(i, u) \quad (14)$$

где $w(i,u)$ – заданная весовая функция, которая оценивает степень важности i-го соседа для классификации объекта u (если гиперпараметр “вес” установлен).

Иллюстрация работы алгоритма KNN представлена на рисунке 30.

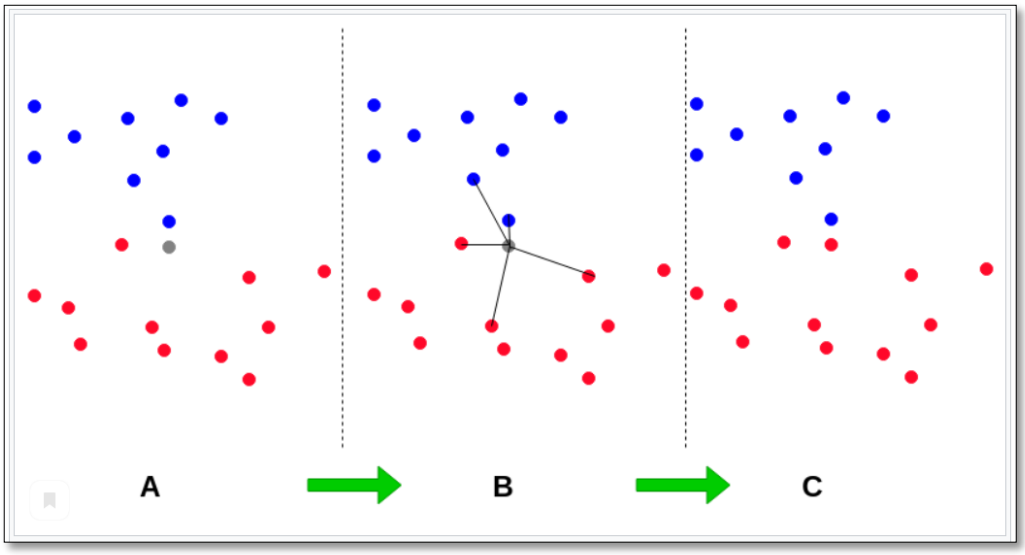


Рисунок 30 – Пример работы алгоритма KNN

Гиперпараметрами для настройки модели являются:

1) K (количество “соседей”);

Основной вопрос заключается в выборе оптимального количества рассматриваемых соседей - небольшое количество соседей создаст слишком большую чувствительность к шумам, в то время как слишком большое количество ухудшит предсказание путем вырождения функции в константу.

2) метрика расстояния;

Выбор наиболее подходящей метрики для измерения расстояния между соседями также может значительно влиять на результат обучения. На рисунке 31 представлены классические метрики и функции для их нахождения.

identifier	class name	args	distance function
“euclidean”	EuclideanDistance	•	`sqrt(sum((x - y)^2))`
“manhattan”	ManhattanDistance	•	`sum(|x - y|)`
“chebyshev”	ChebyshevDistance	•	`max(|x - y|)`

Рисунок 31 – Метрики расстояний для метода KNN

3) веса соседей.

Данная опция позволяет в зависимости от расстояния между исследуемым объектом и его соседями назначать им веса. Чем дальше “сосед” расположен от объекта, тем меньше его вес, и наоборот.

Как и деревья решений, метод используется для бинарной (рисунки 32 - 33) и многоклассовой классификации (рисунок 34).

```
# Реализация метода К-ближайших соседей
from sklearn.neighbors import KNeighborsClassifier # Импорт функции для метода К-ближайших соседей
from sklearn.model_selection import GridSearchCV # Импорт сети параметров для сравнения модели с разными признаками
KNN_Classifier = KNeighborsClassifier() # Объявление классификатора
knn_params = {'n_neighbors': list(range(1,10,1)), 'weights': ['uniform', 'distance'],
              'metric': ['euclidean', 'manhattan', 'chebyshev']} # Набор параметров для метода
knn_grid = GridSearchCV(KNN_Classifier, knn_params) # Заполнение сети моделями со всеми вариантами параметров
knn_grid.fit(x_train_scaled, y_train) # Тренировка моделей
knn_grid.best_score_, knn_grid.best_params_
```

```
(0.7889447236180904,
 {'metric': 'manhattan', 'n_neighbors': 4, 'weights': 'distance'})
```

Рисунок 32 – Обучение модели KNN (детектирование)

Оценим качество предсказаний лучшей модели KNN на тестовой выборке.

```
# Предсказание методом KNN
y_KNN_pred = knn_grid.predict(x_test_scaled)
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test,y_KNN_pred, target_names=['0','1']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           0       0.85      0.89      0.87       296
           1       0.72      0.63      0.67       131

    accuracy                           0.81       427
   macro avg       0.78      0.76      0.77       427
weighted avg       0.81      0.81      0.81       427
```

```
print(y_KNN_pred) # Вывод предсказанных классов для всех объектов тестовой выборки
print(y_KNN_pred[7]) # Вывод предсказанного класса для одного из объектов выборки
```

```
[0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 1 1 0 0 0 0
 0 0 0 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0
 0 0 1 1 1 1 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0
 0 1 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0
 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 1 0 0
 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 0 0 1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0
 1 1 0 0 0 1 1 0 0 0 1 1 1 0 1 1 1 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0
 1 1 1 1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0
 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1
 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0
 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1
 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1]
0
```

Рисунок 33 – Метрики качества и значения предсказаний модели KNN (детектирование)

```
# Предсказание методом KNN
y_KNN_pred = knn_grid.predict(x_test_scaled)
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test,y_KNN_pred, target_names=['1','2','3','4','5','6']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           1       0.57      0.54      0.55        24
           2       0.85      0.91      0.88        45
           3       1.00      0.97      0.98       100
           4       1.00      0.90      0.95        10
           5       0.92      0.86      0.89        14
           6       0.83      0.86      0.84        97

    accuracy                           0.88       290
   macro avg       0.86      0.84      0.85       290
weighted avg       0.88      0.88      0.88       290
```

```
print(y_KNN_pred) # Вывод предсказанных классов для всех объектов тестовой выборки
print(y_KNN_pred[7]) # Вывод предсказанного класса для одного из объектов выборки
```

```
[2 6 6 3 6 2 3 6 4 6 5 3 1 6 6 3 4 3 6 3 4 6 2 6 3 6 6 3 2 2 2 1 6 3 2 6 2
 3 6 5 4 3 3 1 3 3 6 4 3 6 6 1 1 3 6 3 6 3 6 1 6 1 6 6 3 3 6 3 6 6 3 6 2 3
 3 1 1 3 2 2 6 5 6 6 3 2 2 1 6 6 3 2 2 4 3 6 6 6 3 3 3 3 3 2 6 3 3 6 2 4 3
 4 6 3 2 6 6 1 2 2 3 3 3 6 3 6 3 3 6 3 3 5 3 6 2 2 6 6 6 3 1 6 6 3 5 6 2 3
 1 2 3 6 3 6 3 3 1 1 1 6 2 6 6 2 2 6 6 3 6 6 6 2 3 3 2 3 6 2 3 6 2 6 6 6 3
 3 2 1 3 6 1 6 3 1 6 3 6 6 2 3 3 6 6 2 5 3 6 3 6 5 6 6 3 6 5 3 6 3 5 2 2 6
 2 2 3 3 6 6 3 6 6 5 2 6 5 1 3 6 6 2 3 3 2 3 6 6 1 6 3 2 5 3 3 3 6 6 6 3 2
 3 5 6 1 2 6 2 3 2 3 3 3 6 6 3 6 6 6 3 3 2 3 3 3 2 3 3 3 3 1 4]
6
```

Рисунок 34 – Метрики качества и значения предсказаний модели KNN (классификация)

4.4.3 Логистическая регрессия

Логистическая регрессия (англ. logit model) – это статистическая модель, используемая для предсказания вероятности возникновения некоторого события путём подгонки данных к логистической кривой.

Задача формулируется следующим образом. Требуется сконструировать модель $f_{w,b}(x)$, являющуюся линейной комбинацией признаков образца x:

$$f_{w,b}(x) = x * w + b, \tag{15}$$

где w – D-мерный вектор параметров, а b – действительное число.

Однако значения данной функции лежат в области (-∞; +∞), а для осуществления бинарной классификации необходима функция с областью значений (0, 1). Одной из функций, обладающих таким свойством, является стандартная логистическая функция, график которой представлен на рисунке 35.

$$\mathrm{f}(x) = \frac{1}{1+e^{-x}} \tag{16}$$

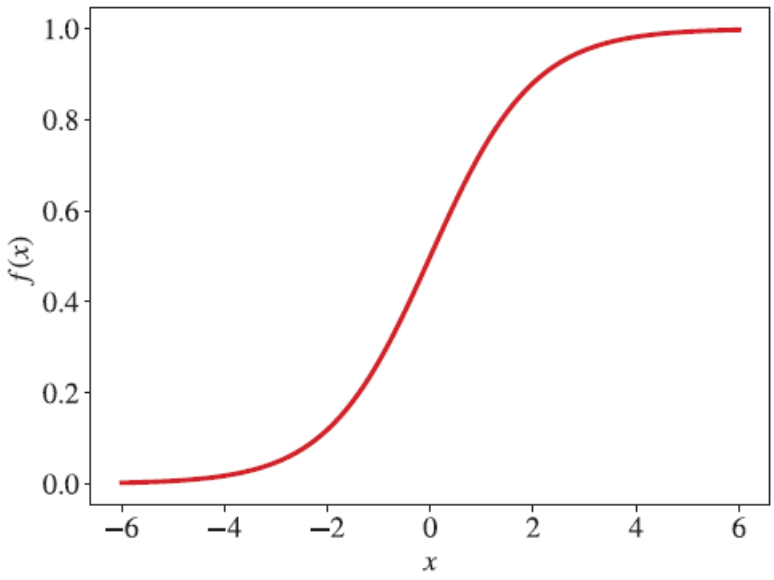


Рисунок 35 – График логистической регрессии

Тогда модель логистической регрессии выглядит так:

$$\mathrm{f}(x) = \frac{1}{1+e^{-(x*w+b)}} \tag{17}$$

Таким образом результат функции (9) можно интерпретировать, как как вероятность, что y_i будет иметь положительное значение. Например, если она выше или равна пороговому значению 0.5, можно утверждать, что класс *x* положителен; иначе – отрицателен.

Так как y принимает лишь значения 0 и 1, то вероятность второго возможного значения:

Для подбора параметров *w* и *b* обычно используется метод максимального правдоподобия, выражающий плотность вероятности (вероятность) совместного появления результатов выборки, согласно которому выбираются параметры, максимизирующие значение функции правдоподобия на обучающей выборке:

$$L_{w,b} = \prod_{i=1..N} f_{w,b}(x_i)^{y_i} * (1 - f_{w,b}(x_i))^{(1-y_i)} \quad (18)$$

Нахождение оценки упрощается, если максимизировать не саму функцию *L*, а натуральный логарифм *ln(L)*, поскольку максимум обеих функций достигается при одинаковых значениях параметра:

$$LogL_{w,b} = \sum_{i=1}^{N} \left[(y_i * ln f_{w,b}(x) + (1 - y_i) * \ln (1 - f_{w,b}(x)) \right] \quad (19)$$

Для максимизации этой функции может быть применён, например, метод градиентного спуска.

Градиент целевой функции (18):

$$\nabla L(w) = \sum_{i=1}^{N} \left(f_{w,b}(x) - y_i \right) * x_i \quad (20)$$

Метод градиентного спуска – итерационный алгоритм минимизации целевой функции, состоящий из следующих шагов:

1) выбирается случайное начальное приближение *w0*;

2) итеративный процесс:

– Приближенное решение w_k на очередной итерации вычисляется как разность между полученным на предыдущей итерации приближенным решением w_{k-1} и вектором градиента $\nabla E(w_{k-1})$, умноженным на коэффициент обучения γ:

$$w_k = w_{k-1} - \gamma \nabla E(w_{k-1}) \quad (21)$$

– Итеративный процесс продолжается до тех пор, пока не будет выполнено следующее условие:

$$\|w_k - w_{k-1}\| < \varepsilon(\|w_k\| + \varepsilon_0), \quad (22)$$

где ε, ε_0 некоторые малые положительных константы.

3) Результат: последнее полученное приближенное решение w_k.

Метод применяется в случае бинарной классификации, однако может применяться и в задачах многоклассовой классификация (так называемый “one-vs-all” метод), заключающийся в приведении искомого класса к одной метке, а всех оставшихся ко второй. Однако для корректного сравнения результатов между различными моделями данный способ решения рассматриваться не будет.

Параметры настройки логистической регрессии:

1) Регуляризация (“penalty”);

Существует несколько типов регуляризации: ‘l1’, ‘l2’, ‘elasticnet’ либо отсутствие регуляризации.

Регуляризация – это метод для уменьшения степени переобучения модели, заключающийся в добавлении еще одного слагаемого в целевую функцию:

$$E_r = E + \frac{\lambda}{2} * \sum_{i=1}^{N} |w_i|^q, \quad (23)$$

где λ – коэффициент регуляризации (англ. regularization coefficient). Выбор показателя степени q зависит от особенностей решаемой задачи, обычно используется значение q = 1 (“l1” или англ. “Lasso”) или q = 2 (“l2” или англ. “Ridge”). “Elasticnet” объединяет оба вышеуказанных метода.

При использовании “Ridge” целевая функция:

$$L_R = \left[\prod_{i=1..N} f_{w,b}(x_i)^{y_i} * (1 - f_{w,b}(x_i))^{(1-y_i)}\right] + \frac{\lambda}{2} * w * w^T \quad (24)$$

Тогда градиент целевой функции:

$$\nabla L(w) = \sum_{i=1}^{N} \left(f_{w,b}(x) - y_i\right) * x_i + \frac{\lambda}{2} \quad (25)$$

Таким образом, использование регуляризации позволяет не допускать очень малых или больших весовых значений для модели.

2) Влияние регуляризации (“C”);

В данном случае в качестве этого параметра выступает коэффициент λ.

3) Вес класса;

”Сбалансированный" режим использует значения *y* для автоматической настройки весов, обратно пропорциональных частотам классов во входных данных в виде:

$$n\_samples / (n\_classes * np.bincount(y)), \quad (26)$$

где n_samples – количество объектов всех классов, n_classes – количество классов, np.bincount(y) – число объектов класса *y*.

4) алгоритм решения (“Solver”).

Для нахождения целевой функции могут быть использованы различные алгоритмы решения (методы на базе градиента и др.), отличающиеся скоростью, возможностями регуляризации и т.д.

Обучение модели детектирование представлено на рисунках 36-38.

```
# Реализация логистической регрессии
from sklearn.linear_model import LogisticRegression # Импорт класса Лог. регр.
from sklearn.model_selection import GridSearchCV # Импорт сети параметров для сравнения модели с разными признаками
Log_reg = LogisticRegression(random_state = 17, n_jobs = -1) # Объявления классификатора
log_reg_params = {"penalty": ['l1', 'l2', 'elasticnet'], "C":np.arange(2,4,0.2), 'class_weight': ['balanced', None],
                  "solver":['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']} # Выбор Solver`a
Log_reg_grid = GridSearchCV(Log_reg, log_reg_params) # Заполнение сети моделями со всеми вариантами параметров
Log_reg_grid.fit(x_train_scaled, y_train) # Обучение всех моделей в сети
```

```
Log_reg_grid.best_score_,  Log_reg_grid.best_params_ # Выдача лучше результата обучения дерева и его параметров
```

```
(0.7547738693467337,
 {'C': 2.0, 'class_weight': None, 'penalty': 'l1', 'solver': 'saga'})
```

Рисунок 36 – Обучение модели “Логистическая регрессия” (детектирование)

```
# Тестирование модели и проверка качества предсказания
from sklearn.metrics import accuracy_score # Импорт метрики "Доля правильных ответов"
y_log_reg_pred = Log_reg_grid.predict(x_test_scaled) # Получение предсказаний модели
accuracy_score(y_log_reg_pred, y_test) # Проверка качества
```

```
0.7704918032786885
```

```
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test, y_log_reg_pred, target_names=['0','1']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           0       0.81      0.88      0.84       296
           1       0.65      0.53      0.59       131

    accuracy                           0.77       427
   macro avg       0.73      0.70      0.71       427
weighted avg       0.76      0.77      0.76       427
```

Рисунок 37 – Метрики качества модели “Логистическая регрессия” (детектирование)

```
# Выдача рез-ов (вероятности принадлежности к классу)
chance_predict = Log_reg_grid.predict_proba(x_test_scaled)*100
```

```
Class_0_chance = chance_predict[:,0]
Class_1_chance = chance_predict[:,1]
```

```
# Вероятности попадания в класс "0" и "1" для трех объектов тестовой выборки
```

```
Class_0_chance[0],Class_0_chance[1],Class_0_chance[2]
```

```
(99.85107171485112, 78.21700771482958, 48.18654335982452)
```

```
Class_1_chance[0],Class_1_chance[1],Class_1_chance[2]
```

```
(0.1489282851488806, 21.78299228517042, 51.81345664017548)
```

Рисунок 38 – Результаты предсказаний модели “Логистическая регрессия” (детектирование)

4.4.4 Наивный Байес

Наивные байесовские алгоритмы – это метод классификации, основанный на применении теоремы Байеса с исходным предположением, состоящее в том, что каждый признак создает независимый и равный вклад результат, иными словами предполагается, что ни одна из пар признаков не является зависимой, а также каждому из признаков присваивается одинаковый вес, т.е он вносит одинаковый вклад в результат.

Теорема Байеса находит вероятность наступления события, учитывая вероятность другого события, которое уже произошло. Математически формулируется следующим образом:

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)}, \quad (27)$$

где $P(B|A)$ – вероятность наступления события B при истинности гипотезы A, $P(A)$ – вероятность события A до наступления события B, $P(B)$ – вероятность события В до наступления события A.

Применительно к имеющемуся набору данных формулировка метода следующая: при наличии переменной класса y и набора признаков $x_1,\ldots,x_n$, верно равенство:

$$P(y|x_1,\ldots,x_n) = \frac{P(x_1,y)*\ldots*P(x_n,y)*P(y)}{P(x_1)*\ldots*P(x_n)} = \frac{P(y)*\prod_{i=1}^{n} P(x_i|y)}{P(x_1)*\ldots*P(x_n)} \quad (28)$$

Для создания модели классификатора необходимо найти вероятность заданного набора входных данных для всех возможных значений переменной класса y и выбираем выходные данные с максимальной вероятностью:

$$y = argmax\, P(y) * \prod_{i=1}^{n} P(x_i|y) \quad (29)$$

Существует несколько вариаций алгоритмов, базирующихся на принципах метода наивного Байеса. Различные алгоритмы отличаются в основном предположениями, которые они делают относительно распределения $P(x \mid y)$.

В работе будет использован гауссовский наивный Байес для идентификации, подразумевающий, что данные для каждого класса имеют гауссово распределение

Также существуют и полиномиальный наивный Байес, где векторы признаков представляют частоты, с которыми определенные события были сгенерированы полиномиальным распределением. Это модель событий, обычно используемая для классификации документов. Еще одним популярным методом является алгоритм Бернулли, где признаками являются независимые логические (двоичные переменные), описывающие входные данные.

Идентификация дефектов методом гауссовского наивного Байеса представлена на рисунках 39 – 41.

```
# Реализация алгоритма наивного Байеса
from sklearn.naive_bayes import GaussianNB # Импорт функции для алгоритма Байес
Bayes_Classifier = GaussianNB() # Объявление классификатора
Bayes_Classifier.fit(x_train_scaled, y_train) # Тренировка модели
```

Рисунок 39 – Обучение модели “Наивный Байес” (детектирование)

```
# Предсказание методом наивного Байеса
y_Bayes_pred = Bayes_Classifier.predict(x_test_scaled)
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test, y_Bayes_pred, target_names=['0','1']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           0       0.88      0.31      0.46       296
           1       0.37      0.91      0.52       131

    accuracy                           0.49       427
   macro avg       0.63      0.61      0.49       427
weighted avg       0.73      0.49      0.48       427
```

Рисунок 40 – Метрики качества для модели “Наивный Байес” (детектирование)

```
# Выдача рез-ов (вероятности принадлежности к классу)
chance_predict = Bayes_Classifier.predict_proba(x_test_scaled)*100
```

```
Class_0_chance = chance_predict[:,0]
Class_1_chance = chance_predict[:,1]
```

```
# Вероятности попадания в класс "0" и "1" для трех объектов тестовой выборки
```

```
Class_0_chance[0],Class_0_chance[1],Class_0_chance[2]
```

```
(100.0, 0.020499291812180652, 0.0007797537272909284)
```

```
Class_1_chance[0],Class_1_chance[1],Class_1_chance[2]
```

```
(3.2231625283459363e-109, 99.97950070818776, 99.99922024627284)
```

Рисунок 41 – Результаты предсказаний для модели “Наивный Байес” (детектирование)

Классификация дефектов методом гауссовского наивного Байеса представлена на рисунках 42 и 43.

```
# Предсказание методом наивного Байеса
y_Bayes_pred = Bayes_Classifier.predict(x_test_scaled)
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test, y_Bayes_pred, target_names=['1','2','3','4','5','6']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           1       0.31      0.92      0.46        24
           2       0.93      0.87      0.90        45
           3       1.00      0.96      0.98       100
           4       1.00      0.90      0.95        10
           5       0.80      0.29      0.42        14
           6       0.87      0.60      0.71        97

    accuracy                           0.79       290
   macro avg       0.82      0.75      0.74       290
weighted avg       0.88      0.79      0.80       290
```

Рисунок 42 – Метрики качества для модели “Наивный Байес” (классификация)

```
# Выдача рез-ов (вероятности принадлежности к классу)
chance_predict = Bayes_Classifier.predict_proba(x_test_scaled)*100

Class_1_chance = chance_predict[:,0]
Class_2_chance = chance_predict[:,1]
Class_3_chance = chance_predict[:,2]
Class_4_chance = chance_predict[:,3]
Class_5_chance = chance_predict[:,4]
Class_6_chance = chance_predict[:,5]

print(chance_predict[0:2,:]) # Вероят-ти принад-ти к классам первой пары объектов
```

```
[[2.04601196e+00 9.79465611e+01 0.00000000e+00 0.00000000e+00
  5.83283992e-07 7.42639850e-03]
 [6.81512951e-02 1.80201448e-05 0.00000000e+00 0.00000000e+00
  8.01666676e-02 9.98516640e+01]]
```

```
print(y_Bayes_pred[0:2]) # Предсказанные классы первой пары объектов
```

```
[2 6]
```

Рисунок 43 – Результаты предсказаний для модели “Наивный Байес” (классификация)

4.4.5 Метод опорных векторов (SVM)

Метод требует, чтобы положительная метка имела числовое значение +1, а отрицательная метка значение –1. SVM рассматривает каждый вектор признаков как точку в многомерном *D* пространстве. Затем алгоритм помещает все векторы признаков на воображаемый *D*-мерный график и рисует воображаемую *D-1*-мерную линию (гиперплоскость), которая отделяет данные с положительными метками от данных с отрицательными метками.

Уравнение гиперплоскости задается двумя параметрами: вещественным вектором w той же размерности, что и входной вектор признаков x, и действительным числом b:

$$f = x * w + b\ , \tag{30}$$

где выражение $x*w$:

$$\mathrm{w} * x = w^{(1)} * x^{(1)} + \cdots + w^{(N)} * x^{(N)}, \tag{31}$$

где N – число измерений.

Тогда математически модель можно выразить так:

$$y = sign(w * x - b) \tag{32}$$

Разделяющую гиперплоскость можно построить разными способами, но в SVM веса и настраиваются таким образом, чтобы объекты классов лежали как можно дальше от разделяющей гиперплоскости. Другими словами, алгоритм максимизирует зазор (англ. margin) между гиперплоскостью и объектами классов, которые расположены ближе всего к ней.

Для нахождения величины ширины между разделяемыми классами необходимо вычислить проекцию вектора w, концами которого будут являться опорные вектора разных классов на вектор w.

Таким образом, необходимо найти расстояние между 2-мя параллельными прямыми (опорными векторами):

$$d = \frac{(-b+1)-(-b-1)}{\sqrt{w^2}} = \frac{2}{\sqrt{w^2}}, \tag{33}$$

где $\sqrt{w^2}$ – евклидова норма $\|w\| = \sqrt{\sum_{i=1}^{D}(w^{(i)})^2}$.

Расстояние между этими гиперплоскостями равно $\frac{2}{\|w\|}$, поэтому задача оптимизации заключается в максимизации этого расстояния, что достигается путем минимизации $\|w\|$.

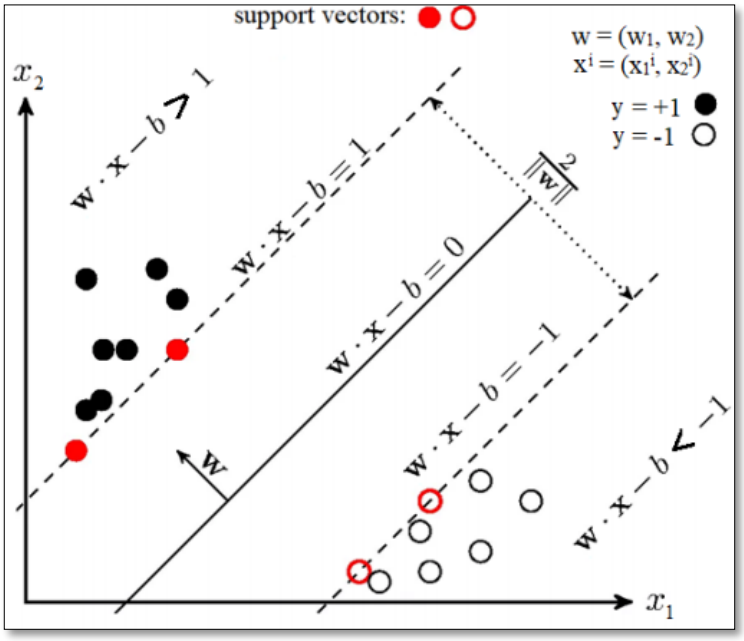


Рисунок 44 – Иллюстрация работы алгоритма “SVM”

Гиперпараметры, настраиваемые при реализации метода “SVM” в Python:

1) шаг регуляризации (“C”);

Гиперпараметр определяет, насколько далеко каждый из элементов в наборе данных имеет влияние при определении “оптимальной линии”. Чем ниже *C*, тем больше элементов, в том числе и те, которые достаточно далеки от разделяющей линии, принимают участие в процессе выбора этой самой линии. Если задать уровень *C* слишком высоким, тогда в процессе принятия решения о расположении линии будут участвовать только самые близкие к линии элементы. Это используется для игнорирования выбросов в данных.

2) ядро функции.

Как было сказано выше, формула (33) используется для описания гиперплоскости в случае линейно разделимых классов. Однако на практике линейно разделимые выборки практически не встречаются: в данных возможны выбросы и нечёткие границы между классами.

В случае линейно-неразделимых классов необходимо преобразовать исходное пространство в пространство более высокой размерности, где данные могут стать линейно разделимыми, т.е используется отображение

$x \to \varphi(x)$, где $\varphi(x)$ – вектор с более высокой размерностью чем x. Например, к двумерным данным, можно применить отображение, проецирующее двумерные данные $x = [q, p]$ в трехмерное пространство $\varphi([q,p]) = (q^2, \sqrt{2}qp, p^2)$.

Однако подобные преобразования со всеми данными вычислительно неэффективны, поэтому используются ядерные функции (или ядра) для эффективной работы в многомерных пространствах без явного преобразования.

Поэтому в случае использования квадратичного ядра вместо преобразования (q_i, p_i) в $(q_i^2, \sqrt{2}q_i p_i, p_i^2)$ и их последующего скалярного произведения используется расчет скалярного произведения (q_i, p_i), чтобы получить $(q_i p_i + q_{i+1} p_{i+1})$ а затем возвести в квадрат, чтобы получить тот же результат.

Одна из самых популярных функций ядра – ядро RBF:

$$k(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right), \quad (34)$$

где $\|x - x'\|$ – евклидово расстояние между 2-мя признаками, σ – гиперпараметр, меняя который можно выбирать между гладкой или изогнутой границей решения в исходном пространстве.

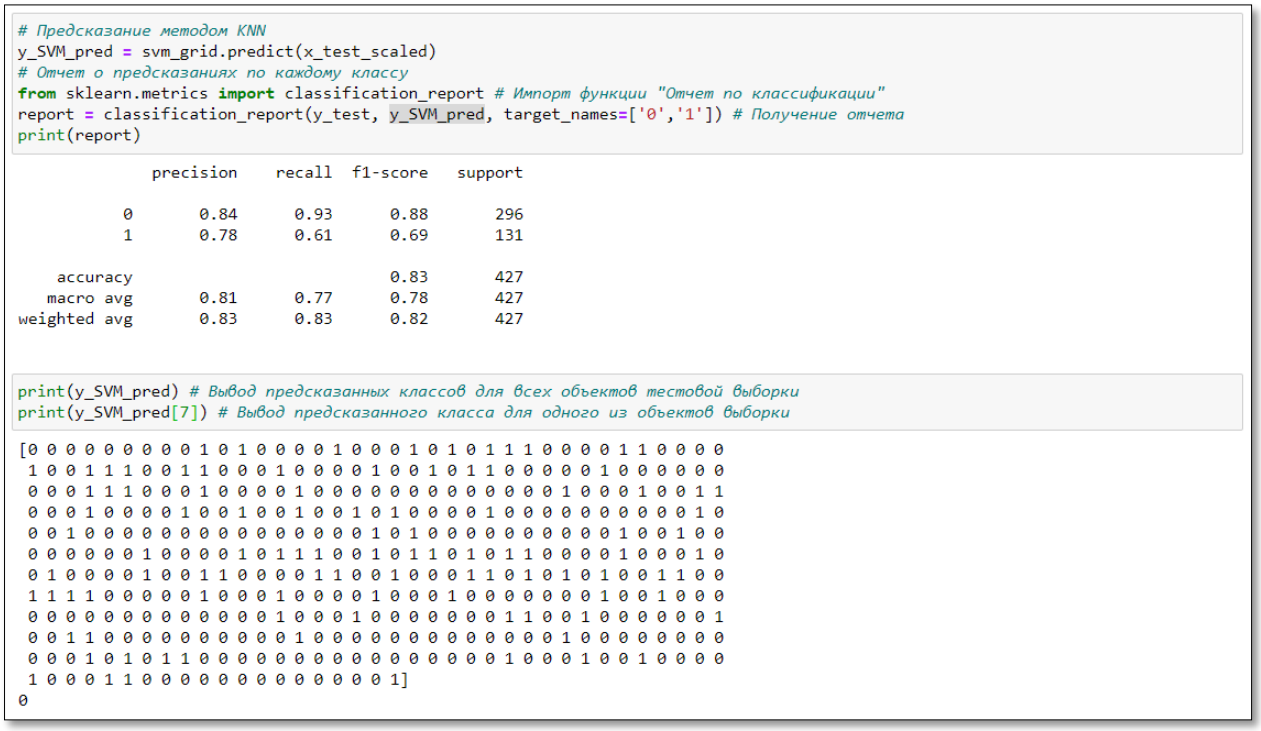
Детектирование дефектов методом “SVM” представлено на рисунках 45 – 46.

```
# Реализация метода К-ближайших соседей
from sklearn import svm # Импорт функции для метода опорных векторов
from sklearn.model_selection import GridSearchCV # Импорт сети параметров для сравнения модели с разными признаками
SVM_Classifier = svm.SVC(random_state = 17) # Объявление классификатора
svm_params = {'C': list(range(0,2,1)), 'kernel': ['linear', 'poly','rbf','sigmoid'],
              'gamma': ['scale', 'auto']} # Набор параметров для метода
svm_grid = GridSearchCV(SVM_Classifier, svm_params) # Заполнение сети моделями со всеми вариантами параметров
svm_grid.fit(x_train_scaled, y_train) # Тренировка моделей
```

```
svm_grid.best_score_, svm_grid.best_params_
```

```
(0.7798994974874371, {'C': 1, 'gamma': 'scale', 'kernel': 'rbf'})
```

Рисунок 45 – Обучение модели “SVM” (детектирование)



```
# Предсказание методом KNN
y_SVM_pred = svm_grid.predict(x_test_scaled)
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test, y_SVM_pred, target_names=['0','1']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           0       0.84      0.93      0.88       296
           1       0.78      0.61      0.69       131

    accuracy                           0.83       427
   macro avg       0.81      0.77      0.78       427
weighted avg       0.83      0.83      0.82       427
```

```
print(y_SVM_pred) # Вывод предсказанных классов для всех объектов тестовой выборки
print(y_SVM_pred[7]) # Вывод предсказанного класса для одного из объектов выборки
```

```
[0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 1 0 1 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0
 1 0 0 1 1 1 0 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0
 0 0 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1
 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0
 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 1 0 0 1 0 1 1 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0
 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 0 1 1 0 1 0 1 0 1 0 0 1 1 0 0
 1 1 1 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1
 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0
 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
0
```

Рисунок 46 – Метрики качества и результаты предсказаний модели “SVM” (детектирование)

4.4.6 Метод XGBoost

Методы, рассмотренные в п. 4.4.1 – 4.4.5, являются классическими алгоритмами машинного обучения. Многие из современных методов, являющиеся комбинированными алгоритмами, базируются на этих моделях.

Основная идея использования комбинированных моделей это усреднение результатов наблюдений может дать более устойчивую и надежную оценку, поскольку ослабляется влияние случайных отклонений в отдельном измерении.

В частности, одним из таких методов является метод экстремального градиентного бустинга.

Бустинг является одним из вариантов реализации подобного способа. Основная идея заключается в последовательном (итеративном) процессе построения частных моделей. Обучение каждой новой модели в этом случае основывается на данных об ошибках предыдущей, где линейная комбинация получившегося ансамбля моделей является результирующей функцией.

Бустинг также имеет несколько разновидностей алгоритмов работы, но в данном разделе рассматривается использование градиентного бустинга – метод подгоняет каждый новый прогнозатор к остаточным ошибкам, допущенным предыдущим прогнозатором [30]. Иными словами, модель развивается в сторону оптимизации целевой функции. Процесс повторяется, пока функция ошибки не перестанет меняться или пока не будет достигнуто максимальное число предикторов.

XGBoost представляет собой хорошо оптимизированный алгоритм градиентного бустинга деревьев решений, имеющий в наличии встроенную регуляризацию, а также возможность задавать пользовательские функции потерь и метрики качества.

Гиперпараметры, настраиваемые при реализации метода XGBoost:

1) “eta”;

Константа (лямбда в формуле)

Определяет степень влияния каждого нового дерева, и таким образом регулирует скорость обучения модели.

2) “gamma”;

Константа, определяющая минимальное уменьшение значения функции потерь.

3) Параметры, относящиеся непосредственно к настройке отдельного дерева.

Описаны в п. 4.4.1.

На рисунке 47 продемонстрировано обучение модели XGBoost, на рисунке 48 отражены результаты тестирования модели для детектирования. Результаты тестирования модели классификации представлены на рисунке 49.

```
# Реализация XGBOOST
import xgboost # Импорт функции для алгоритма XGBoost
from sklearn.model_selection import GridSearchCV # Импорт сети параметров для сравнения модели с разными признаками
XGBoost_Classifier = xgboost.XGBClassifier() # Объявление классификатора
XGBoost_params = {'max_depth': np.arange(3,6), 'eta':[0.3, 0.7, 1],
                  'colsample_bytree':[0.7, 1], 'gamma':[0,0.5]} # Установка диапазанов поиска параметров дерева
XGBoost_grid = GridSearchCV(XGBoost_Classifier, XGBoost_params) # Заполнение сети моделями со всеми вариантами параметров
XGBoost_grid.fit(x_train, y_train) # Обучение всех моделей в сети
```

```
GridSearchCV(cv=None, error_score=nan,
             estimator=XGBClassifier(base_score=None, booster=None,
                                     colsample_bylevel=None,
                                     colsample_bynode=None,
                                     colsample_bytree=None, gamma=None,
                                     gpu_id=None, importance_type='gain',
                                     interaction_constraints=None,
                                     learning_rate=None, max_delta_step=None,
                                     max_depth=None, min_child_weight=None,
                                     missing=nan, monotone_constraints=None,
                                     n_es...
                                     objective='binary:logistic',
                                     random_state=None, reg_alpha=None,
                                     reg_lambda=None, scale_pos_weight=None,
                                     subsample=None, tree_method=None,
                                     validate_parameters=None, verbosity=None),
             iid='deprecated', n_jobs=None,
             param_grid={'colsample_bytree': [0.7, 1], 'eta': [0.3, 0.7, 1],
                         'max_depth': array([3, 4, 5])},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0)
```

```
XGBoost_grid.best_score_,  XGBoost_grid.best_params_ # Выдача лучше результата обучения дерева и его параметров
```

```
(0.8140703517587939, {'colsample_bytree': 1, 'eta': 0.7, 'max_depth': 4})
```

Рисунок 47 – Обучение модели “XGBoost” (детектирование)

```
# Предсказание методом XGBoost
y_XGBoost_pred =XGBoost_grid.predict(x_test)
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test, y_XGBoost_pred, target_names=['0','1']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           0       0.84      0.88      0.86       296
           1       0.70      0.63      0.67       131

    accuracy                           0.81       427
   macro avg       0.77      0.76      0.76       427
weighted avg       0.80      0.81      0.80       427
```

```
print(y_XGBoost_pred)
```

```
[0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 1 0 0 0 0 0
 1 1 0 1 0 1 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1
 0 0 0 1 1 1 0 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1
 0 1 0 0 0 0 0 0 1 0 0 1 0 0 1 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0 1 0 0 1 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 1 1 0 1 1 0 1 0 0 0 1 0 0 1 1 0 0 0 1 0
 1 1 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1 0 0 1 0 0 0 1 1 1 1 0 1 0 1 0 0 1 1 0 0
 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0 0 1
 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0
 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0
 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1]
```

Рисунок 48 – Метрики качества и результаты предсказаний модели “XGBoost” (детектирование)

```
# Предсказание методом XGBoost
y_XGBoost_pred =XGBoost_grid.predict(x_test)
# Отчет о предсказаниях по каждому классу
from sklearn.metrics import classification_report # Импорт функции "Отчет по классификации"
report = classification_report(y_test, y_XGBoost_pred, target_names=['1','2','3','4','5','6']) # Получение отчета
print(report)
```

```
              precision    recall  f1-score   support

           1       0.70      0.58      0.64        24
           2       0.96      0.96      0.96        45
           3       1.00      0.98      0.99       100
           4       1.00      0.80      0.89        10
           5       0.92      0.86      0.89        14
           6       0.84      0.92      0.88        97

    accuracy                           0.91       290
   macro avg       0.90      0.85      0.87       290
weighted avg       0.91      0.91      0.91       290
```

```
print(y_XGBoost_pred)
```

```
[2 6 6 3 3 2 3 6 2 6 5 3 1 6 6 3 4 3 2 3 4 6 2 6 3 1 6 3 2 2 2 6 6 3 2 6 2
 3 6 5 4 3 3 6 3 3 6 4 3 1 6 1 1 3 6 3 6 3 6 1 6 6 6 6 3 3 6 3 6 2 3 6 2 3
 3 1 1 3 2 2 6 5 6 6 3 2 3 2 6 6 3 2 2 4 3 6 6 6 3 3 3 3 3 2 6 3 3 6 6 4 3
 4 6 3 2 6 6 1 2 2 3 3 3 6 3 6 3 3 6 3 3 5 3 6 2 2 6 6 1 3 6 6 6 6 3 5 6 2 3
 6 6 3 6 3 6 3 3 6 1 1 6 6 6 6 2 2 6 6 3 6 6 6 6 2 3 3 2 3 6 6 3 1 2 6 6 6 3
 3 2 1 3 6 1 6 3 1 6 3 6 6 2 3 3 1 6 2 5 3 6 3 6 5 6 6 3 6 5 3 6 3 6 1 2 6
 6 2 6 3 6 6 3 6 6 5 6 6 5 6 3 6 6 2 3 3 2 3 6 6 2 6 3 2 5 3 3 3 6 6 6 3 2
 3 5 6 1 2 6 2 3 2 3 3 3 5 6 3 6 6 6 3 3 2 3 3 3 2 3 3 3 3 1 4]
```

Рисунок 49 – Метрики качества и результаты предсказаний модели “XGBoost” (классификация)

4.4.7 Сравнение результатов

Сравнение результатов тестирования моделей по различным метрикам качества представлено в таблицах 3 и 4.

В качестве общего критерия для сравнения всех моделей будет выбран f1-score, который в условиях несбалансированности классов способен наиболее полно отразить результаты тестирования моделей.

Таблица 3 – Показатели качества моделей в задаче идентификации

Метрика / Модель	**Precision**		**Recall**		**F1-score**		**Среднее (F1)**
	0	**1**	**0**	**1**	**0**	**1**	
KNN	0,85	0,72	0,89	0,63	0,82	0,63	0,73
SVM	0,84	0,78	0,93	0,61	0,88	0,61	0,75
DT	0,83	0,59	0,81	0,62	0,82	0,6	0,71
NB	0,88	0,37	0,31	0,91	0,46	0,52	0,49
LogReg	0,81	0,65	0,88	0,53	0,84	0,59	0,72
XGBoost	0,84	0,70	0,88	0,63	0,86	0,67	0,77

По результатам тестирования моделей наилучшие показатели по метрике f1 показывает метод XGBoost, что подтверждает его статус одного из лучших методов машинного обучения в сравнении с классическими алгоритмами. Также стоит отметить 20% разницу в точности предсказаний между классами с меткой “0” и “1”, которая объясняется дисбалансом классов. Иными словами, количество данных для обучения с меткой “1” меньше, чем с меткой “0”.

Таблица 4 – Показатели качества моделей в задаче классификации

Метод \ Метрика	Precision						Recall						F1-score						$F1_{ср}$
	1	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**	
KNN	0,57	0,85	1,00	1,00	0,92	0,83	0,54	0,91	0,97	0,90	0,86	0,86	0,55	0,88	0,98	0,95	0,89	0,84	0,85
DT	0,50	0,86	0,94	0,78	0,81	0,81	0,58	0,84	0,97	0,70	0,93	0,75	0,54	0,85	0,86	0,74	0,87	0,78	0,77
NB	0,31	0,93	1,00	1,00	0,80	0,87	0,92	0,87	0,96	0,90	0,29	0,60	0,46	0,90	0,98	0,95	0,42	0,71	0,74
XGBoost	0,70	0,96	1,00	1,00	0,92	0,84	0,58	0,96	0,98	0,80	0,86	0,92	0,64	0,96	0,99	0,89	0,89	0,88	0,88

В случае задачи многоклассовой классификации наибольшую точность также показывает метод XGBoost. Стоит отметить более низкую точность в предсказаниях класса “1” у всех моделей в сравнении с другими классами, причина которой может заключаться в недостаточном количестве признаков для этого класса.

Таким образом универсальным средством для решения задач и идентификации, и классификация является метод “XGBoost” с показателями точности 77% и 88% соответственно. Дальнейшее увеличение набора данных позволит увеличить точность предсказаний.

5 Интеграция модели МО в технологический процесс

Непрерывное выполнение оценки качества продукции методами машинного обучения в рамках постоянного протекания технологического процесса обеспечивается выполнением ряда операций.

1) передача данных от средств измерений ОСК в ПЛК;

В условиях отсутствия реального оборудования необходимо смоделировать технологический процесс, чтобы обеспечить поступление данных в виртуальный ПЛК.

В ПО Matlab (Simulink) создается модель, содержащая набор данных, записывающихся в ПЛК, а также специальные блоки для соединения и передачи данных в OPC-сервер. Таким образом данные из модели Matlab передаются в OPC, из которого затем передаются в ПЛК, находящийся в режиме симуляции. В роли OPC-сервера выступает ПО “Process Simulator”, симуляция работы ПЛК осуществляется в среде TIA Portal.

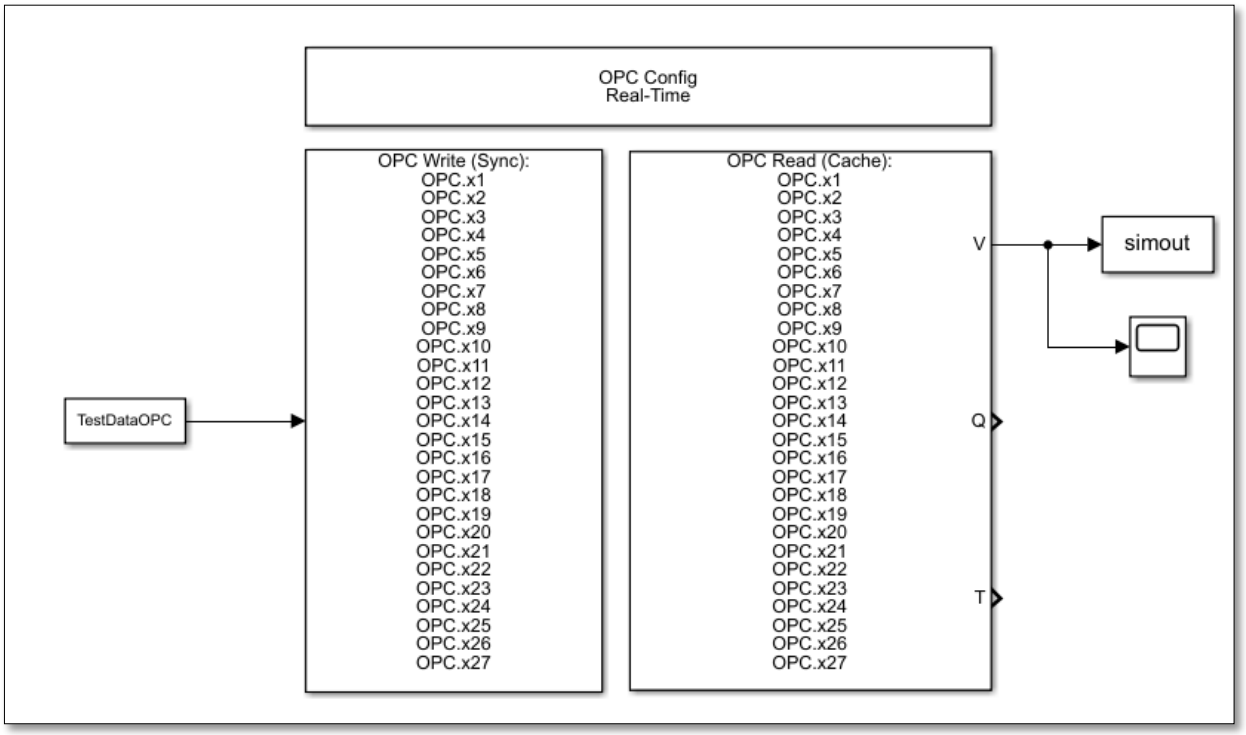


Рисунок 50 – Simulink модель передачи данных в OPC

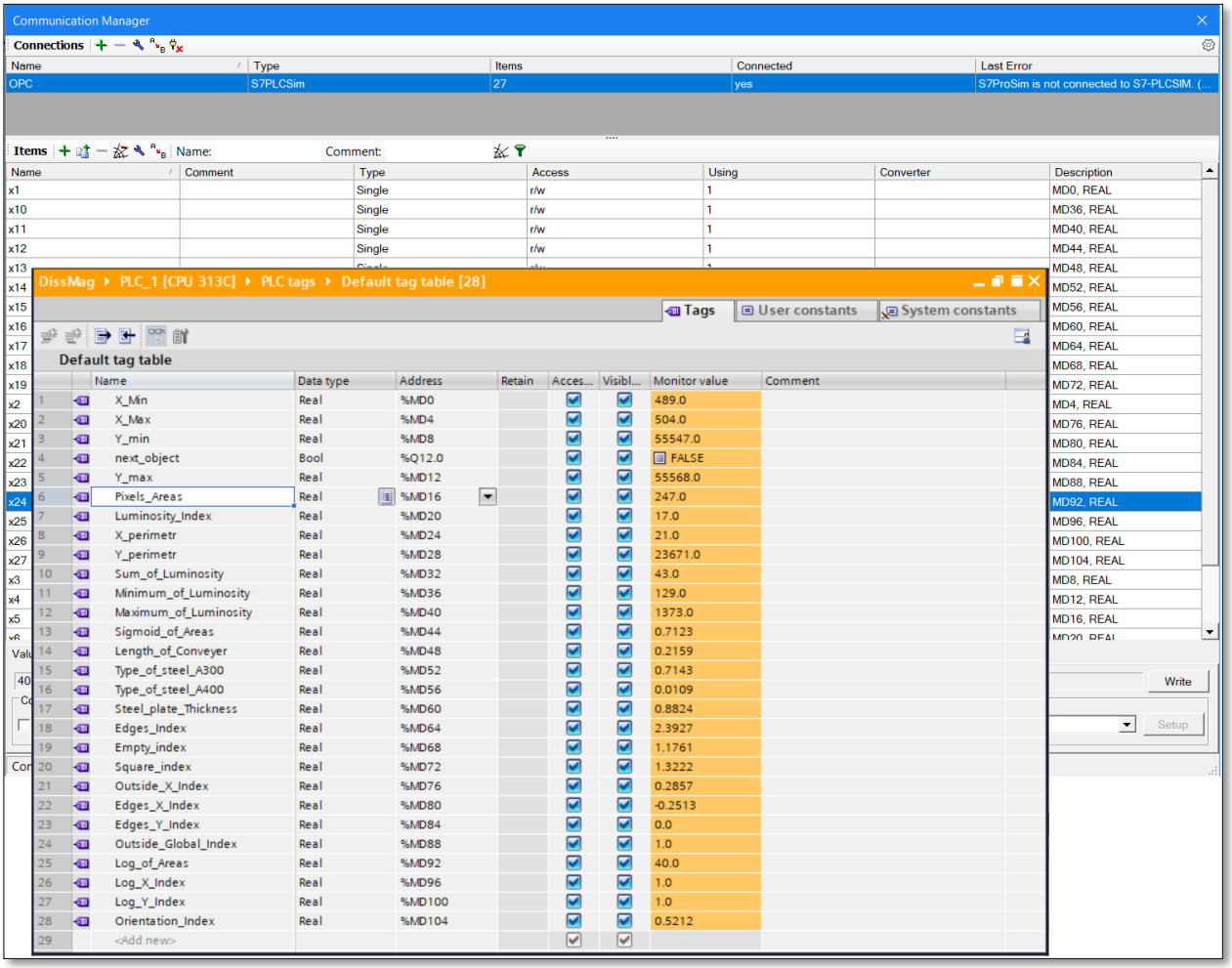


Рисунок 51 – Передача значений из OPC в TIA Portal

2) импортирование значений технологических тегов в среду разработки;

Передача информации из ПЛК в Python может быть выполнена посредством библиотеки Snap7.

```
# Импорт библиотек для работы с набором данных
import numpy as np
import pandas as pd
import XGBoost

# Импорт библиотеки Snap7
import snap7.client as c
from snap7.util import *
from snap7.snap7types import *
# Функция чтения тегов
def ReadMemory(plc,byte,bit,datatype):
    result = plc.read_area(areas['MK'],0,byte,datatype)
    if datatype==S7WLBit:
        return get_bool(result,0,bit)
    elif datatype==S7WLByte or datatype==S7WLWord:
        return get_int(result,0)
    elif datatype==S7WLReal:
        return get_real(result,0)
    elif datatype==S7WLDWord:
        return get_dword(result,0)
    else:
        return None

# Соединение с ПЛК
if __name__=="__main__":
    plc = c.Client()
    plc.connect('192.168.0.1',0,1)

# Чтение тегов и запись значений в массив
adr = 0
tags = np.array
for adr in Adress:
    np.append(tags,ReadMemory(plc,adr,0,S7WLReal))
print(tags)

[ 4.8900e+02  5.0400e+02  5.5547e+04  5.5568e+04  2.4700e+02  1.7000e+01
  2.1000e+01  2.3671e+04  4.3000e+01  1.2900e+02  1.3730e+03  7.1000e-01
  2.2000e-01  7.1000e-01  1.0000e-02  8.8000e-01  2.3900e+00  1.1800e+00
  1.3200e+00  2.9000e-01 -2.5000e-01  0.0000e+00  1.0000e+00  4.0000e+01
  1.0000e+00  1.0000e+00  5.2000e-01]
```

Рисунок 52 – Импортирование значений тегов из ПЛК в Python

Оставшиеся операции заключаются в приведении массива данных в формат “DataFrame” и получения результата с помощью выбранной и обученной ранее модели.

3) Запись данных измерений ОСК в БД;

Помимо передачи измерений в ПЛК все параметры заготовки также сохраняются в БД. Это необходимо как для архивирования данных, так и для их использования в целях обучения моделей по мере накопления информации о дефектах.

Поскольку значения всех признаков заготовок необходимы в первую очередь для специалистов МО, то для их хранения могут подойти реляционные база данных, способные хранить огромные массивы информации. В данном случаем высокая скорость обработки информации и индексирование по времени не играет принципиальной роли, поэтому использование БД временных рядов не имеет смысла.

4) Обучение модели.

Модели могут обучаться с определенной частотой, определяемой экспертами МО на производстве, или по получении установленного количества новых данных.

Таким образом набором данных для обучения будет исходный набор данных с добавлением новых данных, в результате чего должны быть выполнены ранее описанные методы по обработке информации.

6 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Научно-исследовательская работа направлена на разработку программного модуля, способного осуществлять предсказания касательно наличия дефекта в конечной продукции, а также тип этого дефекта. В работе рассматривается применение проектируемого модуля в качестве инструмента, способного производить первичную оценку качества выпускаемой продукции (в данном случае металлических заготовок).

Целью данной главы является определение потенциала разрабатываемого модуля, планирование процесса управления исследования, определение ресурсной, финансовой и экономической эффективности.

Достижение цели обеспечивается решением следу задач:

1) Организация работ по разрабатываемому проекту;
2) Планирование работ по проекту;
3) Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования.

6.1 Организация и планирование работ

Занятость каждого из участников при проведении каждого из этапов работ должна быть оптимально спланирована по срокам. На каждом этапе выполнения работ определяются исполнители и продолжительность каждого этапа. Календарный график реализации проекта – это результат планирования работ. Он представляет из себя наиболее наглядный и удобный способ организации проекта. Для его построения определяются даты начала и окончания работ, а также составляется перечень работ и соответствие работ своим исполнителям. Основные исполнители в проекте: научный руководитель (НР) и инженер (И).

Разделение выполнения дипломной работы на этапы представлены в таблице 5.

Таблица 5 – Перечень работ и продолжительность их выполнения

Этапы работы	**Исполнители**	**Загрузка исполнителей**
Определение темы ВКР	НР, И	НР – 100% И – 50%
Поиск и изучение нормативно-технической литературы	НР, И	НР – 30% И – 100%
Календарное планирование работ	НР, И	НР – 100% И – 10%
Сбор данных с технологического процесса	И	И – 100%
Описание технологического процесса	И	И – 100%
Анализ методов машинного обучения	И	И – 100%
Обработка данных для обучения моделей	И	И – 100%
Математическое описание используемых моделей	НР, И	НР – 30% И – 100%
Обучение и подбор оптимальных параметров для каждой из моделей	И	И – 100%
Тестирование моделей	И	И – 100%
Сравнение результатов работы моделей	И	И – 100%
Написание раздела «финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	И	И – 100%
Написание раздела «социальной ответственности»	И	И – 100%
Проверка работы с руководителем	НР, И	НР – 100% И – 10%
Составление пояснительной записки	НР, И	НР – 30% И – 100%
Подготовка презентации и выступления для защиты дипломного проекта	И	И – 100%

6.1.1 Продолжительность этапов работ

Для определения ожидаемого (среднего) значения трудоемкости используется следующая формула:

$$t_{ож} = \frac{3*t_{min\,i}+2*t_{max\,i}}{5}, \quad (35)$$

где t_{min} – минимальная трудоемкость i-ой работы, чел/дн.; t_{max} – максимальная трудоемкость i-ой работы, чел/дн.

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести ее в календарные дни. Расчет продолжительности выполнения каждого этапа в рабочих днях ($T_{РД}$) ведется по формуле:

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д}, \quad (36)$$

где $t_{ож}$ – продолжительность работы, дн., $K_{вн}$ – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей ($K_{вн}$ = 1); $K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ($K_{Д}$ =1,2).

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{РД} \cdot T_{К}, \quad (37)$$

где $T_{КД}$ – продолжительность выполнения этапа в календарных днях, $T_{К}$ – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях, и рассчитываемый по формуле:

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ}-T_{ВД}-T_{ПД}}, \quad (38)$$

где $T_{КАЛ}$ – календарные дни, $T_{ВД}$ – выходные дни, $T_{ПД}$ – праздничные дни.

Рассчитаем коэффициент календарности для пятидневной рабочей недели из расчета 145 нерабочих дней на 2020 год:

$$T_{к} = \frac{T_{кал}}{T_{кал}-T_{вых}-T_{пр}} = \frac{365}{365-118} = 1{,}47. \quad (39)$$

В таблице 6 приведены продолжительности этапов работ и их трудоемкости по исполнителям, занятым на каждом этапе. Стоит отметить, что величины трудоемкости этапов по исполнителям $T_{КД}$ позволяют построить линейный график осуществления проекта, представленный на рисунке 1.

Таблица 6 – Трудозатраты на выполнение проекта

Этап	**Исполнители**	**Продолжительность работ, дни**			**Трудоемкость работ по исполнителям чел.- дн.**			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	**НР**	**И**	**НР**	**И**
Определение темы ВКР	НР, И	1	3	1,8	2,2	1,1	3,2	1,6
Поиск и изучение нормативно-технической литературы	НР, И	4	7	5,2	1,9	6,2	2,8	9,2
Календарное планирование работ	НР, И	1	2	1,4	1,7	0,2	2,5	0,2
Сбор данных с технологического процесса	И	2	3	2,4	0,0	2,9	0,0	4,2
Описание технологического процесса	И	4	6	4,8	0,0	5,8	0,0	8,5
Анализ методов машинного обучения	И	4	5	4,4	0,0	5,3	0,0	7,8
Обработка данных для обучения моделей	И	6	9	7,2	0,0	8,6	0,0	12,7
Математическое описание используемых моделей	НР, И	8	10	8,8	3,2	10,6	4,7	15,5
Обучение и подбор оптимальных параметров для каждой из моделей	И	6	10	7,6	0,0	9,1	0,0	13,4
Тестирование моделей	И	3	5	3,8	0,0	4,6	0,0	6,7
Сравнение результатов работы моделей	И	3	5	3,8	0,0	4,6	0,0	6,7

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.- дн.			
					$T_{РД}$		$T_{КД}$	
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
Написание раздела «финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	И	5	7	5,8	0,0	7,0	0,0	10,2
Написание раздела «социальной ответственности»	И	4	6	4,8	0,0	5,8	0,0	8,5
Проверка работы с руководителем	НР, И	3	4	3,4	4,1	1,2	6,0	1,8
Составление пояснительной записки	НР, И	2	3	2,4	0,9	2,9	1,3	4,2
Подготовка презентации и выступления для защиты дипломного проекта	И	4	6	4,8	0,0	5,8	0,0	8,5
Итого				72,4	13,8	81,4	20,3	119,7

На основе календарного плана проекта построена диаграмма Ганта (рисунок 53), которая наглядно показывает следование выполнения этапов дипломного проектирования, исходя из отведенных сроков. Диаграмма Ганта – это тип столбчатых диаграмм, который используется для иллюстрации календарного плана проекта, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

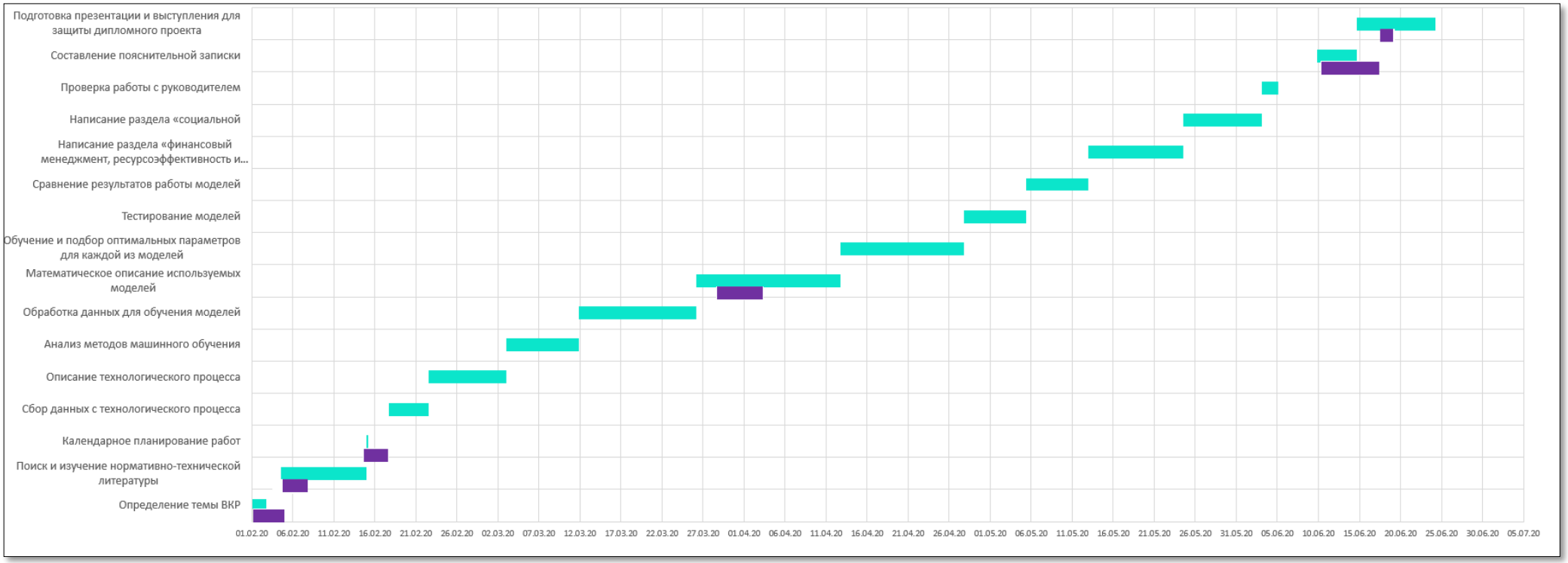


Рисунок 53 – Линейный график осуществления проекта

6.2 Расчет сметы затрат на выполнение проекта

При планировании бюджета исследования должно быть обеспечено полное и достоверное отражение всех видов планируемых расходов, необходимых для его выполнения.

В состав затрат на создание проекта включается величина всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Расчет сметной стоимости ее выполнения производится по следующим статьям затрат:

1) материалы и покупные изделия;
2) заработная плата;
3) социальный налог;
4) расходы на электроэнергию (без освещения);
5) амортизационные отчисления;
6) командировочные расходы;
7) оплата услуг связи;
8) арендная плата за пользование имуществом;
9) прочие услуги (сторонних организаций);
10) прочие (накладные) расходы.

6.2.1 Затраты на материалы и покупные изделия

К данной статье расходов относится стоимость материалов, покупных изделий, полуфабрикатов и других материальных ценностей, расходуемых непосредственно в процессе выполнения работ над объектом проектирования. Кроме того, статья включает так называемые транспортно-заготовительные расходы.

Для выполнения расчетов расходы принимаются как 15 % от отпускной цены закупаемых материалов. Затраты на материалы сведены в таблицу 7.

Таблица 7 – Расчет затрат на материалы

Наименование материалов	Цена за ед., руб.	Кол-во	Сумма, руб.
OPC-сервер Lectus	4 800	1 экз.	4 800
Microsoft Office 2016 (лицензионное ПО)	3400	1 шт.	3400
Лицензионное ПО MatLaB	5600	1 экз.	5600
Бумага для принтера формата А4	250	1 уп.	250
Картридж для принтера	1800	1 шт.	1800
Итого			15850

Расходы на материалы с учетом ТЗР составляют:

$$С_{мат} = 15850 * 1,15 = 18277,5 \text{ руб.} \quad (40)$$

6.2.2 Затраты на заработную плату

Данная статья расходов включает заработную плату работников, непосредственно занятых выполнением НТИ (в том числе премии, доплаты и т.д)

Среднедневная тарифная заработная плата ($ЗП_{дн-т}$) рассчитывается по формуле:

$$ЗП_{дн-т} = МО/20,6 \quad (41)$$

учитывающей, что в году 247 рабочих дней (на 2020 год) и, следовательно, в месяце в среднем 20,6 рабочих дня (при пятидневной рабочей неделе).

Расчеты затрат на полную заработную плату приведены в таблице 8. Затраты времени по каждому исполнителю в рабочих днях с округлением до целого взяты из таблицы 6.

Для учета в ее составе премий, дополнительной зарплаты и районной надбавки используется следующие коэффициенты (с учетом 5-ти дневной рабочей недели): $К_{ПР}$ = 1,1; $К_{доп.ЗП}$ = 1,113 (используется для 6-ти дневной рабочей недели); $К_р$ = 1,3. Таким образом, для перехода от тарифной (базовой) суммы заработка исполнителя, связанной с участием в проекте, к

соответствующему полному заработку (зарплатной части сметы) необходимо первую умножить на интегральный коэффициент:

$$К_и = 1,1* 1,118*1,3 = 1,6. \quad (42)$$

Таблица 8 – Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб.день	Затраты времени, раб.дни	Коэффициент	Фонд з/п., руб.
НР	33 664	1634	14	1,6	36601,6
И	13 500	655	82	1,6	85936
Итого					122537,6

6.2.3 Затраты на социальный налог

Затраты на единый социальный налог (ЕСН), включающий в себя отчисления в пенсионный фонд, на социальное и медицинское страхование, составляют 30% от полной заработной платы по проекту, т.е. $С_{соц.} = С_{зп}*0,3$. Тогда расходы на социальный налог составляют:

$$С_{соц.} = 122537,6 * 0,3 = 36761,28 \text{ руб.} \quad (43)$$

6.2.4 Затраты на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, потраченную при выполнении проекта на работу используемого оборудования, рассчитываемые по формуле:

$$С_{эл.об.} = Р_{об} \cdot t_{об} \cdot Ц_Э, \quad (44)$$

где $P_{ОБ}$ – мощность, потребляемая оборудованием, кВт, $Ц_Э$ – тариф на 1 кВт·час, $t_{об}$ – время работы оборудования, час.

Для ТПУ $Ц_Э$ = 6,59 руб./квт·час (с НДС).

Время работы оборудования вычисляется на основе затраченного времени (см. таблицу 6) для инженера ($T_{РД}$) из расчета 8-ми часового рабочего дня:

$$t_{об} = T_{РД}* К_t, \quad (45)$$

где $К_t \leq 1$– коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к $T_{РД}$, примем данный коэффициент равным 0,8.

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{ОБ} = P_{ном.} * К_С, \quad (46)$$

где $P_{ном.}$ – номинальная мощность оборудования, кВт, $K_С \leq 1$ – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности $K_С = 1$.

Затраты на электроэнергию для технологических целей приведен в таблице 9.

Таблица 9 – Затраты на электроэнергию технологическую

Наименование оборудования	**Время работы оборудования $t_{об}$, час**	**Потребляемая мощность $Р_{ОБ}$, кВт**	**Затраты $Э_{ОБ}$, руб.**
Персональный компьютер	81,4*8*0,8 = 520,96	0,3	1029,93
Лазерный принтер	2	0,1	1,32
Итого			1031,25

6.2.5 Амортизационные расходы

В статье «Амортизационные отчисления» рассчитывается амортизация используемого оборудования за время выполнения проекта.

Используется формула:

$$С_{АМ} = \frac{Н_А * Ц_{ОБ} * t_{рф} * n}{F_Д}, \quad (47)$$

где $Н_А$ – годовая норма амортизации единицы оборудования, $Ц_{ОБ}$ – балансовая стоимость единицы оборудования с учетом ТЗР, $F_Д$ – действительный годовой фонд времени работы соответствующего оборудования, берется из фактического режима его использования в текущем календарном году, $t_{рф}$ –

фактическое время работы оборудования в ходе выполнения проекта, *n* – число задействованных однотипных единиц оборудования.

Определение $Н_А$ содержится в постановлении правительства РФ «О классификации основных средств, включенных в амортизационные группы», согласно которому граничные значения сроков амортизации (полезного использования) оборудования (для ПК сроки составляют 2 ÷ 3 года). Для расчетов принимается значение 2 года. Значение величины $Н_А$ обратно пропорционально значению сроку амортизации ($Н_А$ = 0,5). Для лазерного принтера принимается аналогичное значение.

Расчет $С_{АМ}$ для ПК выглядит следующим образом. Стоимость ПК 70000 руб., время использования 654,4 часа (см. таблицу 6), $Н_А$ = 0,5, $F_Д$ = 220 * 8 = 1760 часов (для пятидневной рабочей недели):

$$С_{АМ} = \frac{0{,}5 * 70000 * 654{,}4 * 1}{1760} = 13013{,}63 \text{ руб.} \quad (48)$$

Значения параметров для принтера. Стоимость принтера 10000 руб., время использования 2 часа, $Н_А$ = 0,5, $F_Д$ = 440 часов:

$$С_{АМ} = \frac{0{,}5 * 10000 * 2 * 1}{440} = 22{,}72 \text{ руб.} \quad (49)$$

Итого начислено амортизации 13036,35 руб.

6.2.6 Расходы, учитываемые непосредственно на основе платежных документов

Здесь учитывается оплата услуги интернет-связи за 5 месяцев (с 01.02 по 01.06) при ежемесячной плате 330 рублей. Таким образом, $С_{нр1}$ = 330 * 5 = 1650 руб.

Также к данному разделу расходов относится и подписка на платные интернет-издания, публикующие научные статьи. За аналогичный период времени $С_{нр2}$ = 180 * 5 = 900 руб.

Таким образом $С_{нр} = С_{нр1} + С_{нр2}$ = 2350 руб.

6.2.7 Прочие расходы

В статье «Прочие расходы» отражены расходы на выполнение проекта, которые не учтены в предыдущих статьях (принимаются равными 10 % от суммы всех вышеуказанных расходов):

$$С_{проч.} = (С_{мат} + С_{зп} + С_{соц} + С_{эл.об.} + С_{ам} + С_{нп}) \cdot 0,1 =$$

$$(18277 + 122537 + 36761,28 + 1031 + 13036 + 2350) \cdot 0,1 = 19399 \text{ руб.} \quad (50)$$

6.2.8 Общая себестоимость разработки

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта.

$$С_{проч.}=С_{мат}+С_{зп}+С_{соц}+С_{эл.об.}+С_{ам}+С_{нп}+С_{п} = 193990+19399=213389 \quad (51)$$

6.2.9 Доход

Поскольку работа заключается в проведении исследования, результаты которого могут использоваться в различных предприятиях, условиях и т.д, то для получения дохода выручка с проекта должна превышать себестоимость (для расчета принимается 10 % превышение от полной себестоимости проекта).

Таким образом доход составляет 21338,9 руб.

6.2.10 Затраты на НДС

На 2020 год НДС в РФ составляет 20% от суммы затрат на разработку и дохода. Тогда затраты на НДС составляю

$$Ц_{НДС} = (213389 + 21338,9) * 0,2 = 46945,58 \text{ руб.} \quad (52)$$

6.2.11 Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС:

$$Ц_{НИР(КР)} = 213389 + 21338,9 + 46945,58 = 281673,48 \text{ руб.} \quad (53)$$

6.3 Оценка экономической эффективности проекта

Основная цель проекта заключается в исследовании методов машинного обучения и создания на их основе программного модуля для решения задач идентификации и классификации дефектов в конечной продукции.

Проведение операций по контролю качества человеком неизбежно влечет за собой ошибки в определении наличия дефекта и его правильной классификации.

Внедрение программного модуля позволит на начальном этапе (в условиях относительно небольшого наличия исходных данных) выступать ему в качестве помощника (экспертной системы) для сотрудников, осуществляющих контроль качества.

Однако дальнейшая работа модуля по накоплению статистических данных технологического процесса и его обучения позволит предприятию заменить автоматизированную операцию по определению качества на автоматическую, что позволит сэкономить средства на заработную плату соответствующим сотрудникам, а также повысить скорость и качество выполняемых операций. Количественная оценка ожидаемого экономического эффекта и соответственно эффективности внедрения разработки в рамках ВКР невозможна.

7 Социальная ответственность

Введение

В данном разделе ВКР будет проведен анализ вредных и опасных факторов, которые могут оказывать влияние как на разработчика ПО, так и на работу персонала, в частности на оператора СУУТП, в связи с внедрением на производство программного модуля (ПМ). Данный программный модуль может использоваться в технологической деятельности промышленных предприятий, лабораторий и т.д, нуждающихся в контроле качества продукции.

В рамках данного раздела будет рассмотрено рабочее место разработчика, которое оборудовано необходимой техникой в связи с разработкой и внедрением ПМ.

Также будут разработаны меры по защите и снижению негативного влияния производственных факторов для рабочего места оператора согласно требованиям, а также даны рекомендации для создания благоприятных условий труда и охраны окружающей среды.

При работе с разрабатываемым модулем человек подвергается различным воздействиям таким как:

1) Отклонения значений температуры и влажности от нормы;
2) Недостаточная освещенность;
3) Повышенный уровень шума и вибрации;
4) Повышенный уровень электромагнитного излучения;
5) Поражение током.

7.1 Аннотация

Написание выпускной квалификационной работы осуществлялось в компании АО «ТомскНИПИнефть» в рамках преддипломной практики, на рабочем месте инженера отела АСУТП.

Поскольку одним из главных векторов направления развития АО “ТомскНИПИнефть” является цифровизация производства, то в компании активно обсуждаются и уже начинают внедряться технологии, использующие нейронные сети, алгоритмы машинного обучения, предназначенные для обработки данных, осуществления предиктивного анализа и т.д.

Поэтому в рамках прохождения практики осуществлялась следующие задачи:

1) Найти и определить исходный набор данных, необходимый для обучения;
2) Сформировать тренировочный и тестовый наборы данных;
3) Определить набор методов (модели), с помощью которых будет происходить обучение;
4) Обучить модели на тренировочном наборе данных и сравнить результаты;
5) Определить наилучшие параметры для каждой из моделей;
6) Сравнить точность моделей на тестовом наборе данных;
7) Создание программного модуля на базе разработанных моделей машинного обучения.

Данный модуль может применятся в рамках ПО, устанавливаемого на АРМ оператора СУУТП любого из предприятий, нуждающихся в автоматизированном контроле качества конечной продукции.

Разрабатываемый программный модуль представляет собой алгоритм расчета вероятности наличия дефекта и принадлежности его к определенному классу на основе получения телеметрической информации с датчиков, установленных на конвейерной ленте в специальной камере.

В связи с внедрением ПМ разработчик работает с таким оборудованием как ПЭВМ и измерительные устройства (датчики).

Помещение, в котором осуществлялась работа по созданию модуля, является офисным кабинетом, где длина – 7 м, ширина – 3 м, высота – 4 м, площадь – 21 м^2, объем – 84 м^3, освещение – естественное и искусственное.

7.2 Правовые и организационные вопросы обеспечения безопасности

Согласно ТК РФ, N 197-ФЗ каждый работник имеет право на:

1) рабочее место, соответствующее требованиям охраны труда;

2) обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний в соответствии с федеральным законом;

3) отказ от выполнения работ в случае возникновения опасности для его жизни и здоровья вследствие нарушения требований охраны труда, за исключением случаев, предусмотренных федеральными законами, до устранения такой опасности;

4) обеспечение средствами индивидуальной и коллективной защиты в соответствии с требованиями охраны труда за счет средств работодателя;

5) внеочередной медицинский осмотр в соответствии с медицинскими рекомендациями с сохранением за ним места работы (должности) и среднего заработка во время прохождения указанного медицинского осмотра.

Разработка модуля подразумевала работу 40 часов в неделю, что соответствует 5-дневной рабочей недели с продолжительностью дня не более 8-ми часов. Согласно 108 ТК РФ [35], работнику предоставляется перерыв для отдыха или питания. Перерыв в организации составляет 45 минут (при максимальной продолжительности два часа).

Рабочее место должно соответствовать требованиям ГОСТ 12.2.032-78 [36]. Оно должно занимать площадь не менее 6 м², высота помещения должна быть не менее 4 м, а объем - не менее 20 м^3 на одного человека. Высота над уровнем пола рабочей поверхности, за которой работает оператор, должна

составлять 720 мм. Оптимальные размеры поверхности стола 1600 х 1000 кв. мм.

Под столом должно иметься пространство для ног с размерами по глубине 650 мм. Рабочий стол должен также иметь подставку для ног, расположенную под углом 15° к поверхности стола. Длина подставки 400 мм, 124 ширина - 350 мм. Удаленность клавиатуры от края стола должна быть не более 300 мм, что обеспечит удобную опору для предплечий.

Расстояние между глазами оператора и экраном видеодисплея должно составлять 40 - 80 см.

Так же рабочий стол должен быть устойчивым, иметь однотонное неметаллическое покрытие, не обладающее способностью накапливать статическое электричество.

Рабочий стул должен иметь дизайн, исключающий онемение тела из-за нарушения кровообращения при продолжительной работе на рабочем месте.

7.3 Производственная безопасность

Программный модуль подразумевает использование ПК и серверного оборудования, работающих в режиме реального времени ежедневно и круглосуточно. С точки зрения социальной ответственности целесообразно рассмотреть вредные и опасные факторы, которые могут возникать при разработке программного модуля или работе с серверным оборудованием, а также требования по организации рабочего места.

Фактор, воздействие которого на работающего в определенных условиях может привести к заболеванию, снижению работоспособности и (или) отрицательному влиянию на здоровье потомства, является вредным. Опасный производственный фактор – фактор, воздействие которого на работающего в определенных условиях приводит к травме, острому отравлению или другому внезапному резкому ухудшению здоровья, или смерти.

Выбор факторов обуславливается документом ГОСТ 12.0.003-2015 «Опасные и вредные производственные факторы. Классификация» [37]. Перечень опасных и вредных факторов, характерных для проектируемой производственной среды представлен в виде таблицы 10.

Таблица 10 – Опасные и вредные факторы при выполнении работ по разработке программного модуля

Факторы (ГОСТ 12.0.003-2015)	**Этапы работ**		**Нормативные документы**
	Разработка	**Эксплуатация**	
Отклонение показателей микроклимата	+	+	СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений
Недостаточная освещенность рабочей зоны	+	+	СНиП 23-05-95* Естественное и искусственное освещение
Повышенный уровень шума	+	+	ГОСТ 12.1.003-2014 Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности
Электромагнитные излучения	+	+	СанПиН 2.2.4.1191-03 Электромагнитные поля в производственных условиях.
Поражение электрическим током	+	+	ГОСТ 12.2.007.0-75 ССБТ. Изделия электротехнические. Общие требования безопасности (с Изменениями N 1, 2, 3, 4)
Опасные факторы, связанные с пожаром	+	+	ГОСТ 12.1.004-91 ССБТ. Пожарная безопасность. Общие требования.

7.4 Анализ вредных и опасных факторов

7.4.1 Электромагнитные излучения

При разработке и использовании программного модуля основным источником опасных факторов является компьютерная техника (ЭВМ), а именно серверное оборудование, а также электрический ток, являющийся источником питания для оборудования. Использование данного оборудования может привести к возникновению таких вредных факторов, как повышенный

уровень статического электричества, повышенный уровень электромагнитных полей, повышенная напряженность электрического поля.

К основной документации, которая регламентирует вышеперечисленные вредные факторы относится [31]. ЭВМ должны соответствовать требованиям настоящих санитарных правил и каждый их тип подлежит санитарно-эпидемиологической экспертизе с оценкой в испытательных лабораториях, аккредитованных в установленном порядке. Допустимые уровни электромагнитных полей (ЭМП), создаваемых ЭВМ, не должны превышать значений [31], представленных в таблице 11.

Таблица 11 – Допустимые уровни ЭМП, создаваемых ЭВМ

Наименование параметров	Диапазон	ДУ ЭМП
Напряженность электрического поля	в диапазоне частот 5 Гц - 2 кГц	25 В/м
	в диапазоне частот 2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц - 400 кГц	25нТл
Поверхностный электростатический потенциал экрана видеомонитора		500В

Допустимое время пребывания работника на рабочем месте (п.7.2.3 в источнике [38]) рассчитывается по формуле:

$$t = \frac{50}{E} - 2, \quad (54)$$

где E – уровень напряженности эл. поля (кВ/м).

Работа по созданию ПМ производилась без использования средств индивидуальной защиты (СИЗ), следовательно параметр E составляет не более 5 кВ/м. Тогда допустимое время равно:

$$t = \frac{50}{5} - 2 = 8 \text{ ч}$$

Результат не превосходит длительности рабочего дня, а значит допустимое время соответствует нормам.

Требования к электрической безопасности при работе на ЭВМ:

1) Для предотвращения поражения электрическим током помещения, где размещаются рабочие места с ЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации.

2) Не следует размещать рабочие места с ЭВМ вблизи силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе ЭВМ. [32]

Согласно разделу 1.1.13 правил устройства электроустановок (ПУЭ) [39] классификация помещений по степени опасности поражения электрическим подразумевает разделять помещения на три отдельных категории, характеризующих степень опасности:

1) особо опасные;

2) с повышенной опасностью;

3) без повышенной опасности.

Помещение, предназначенное для исследования и использования результатов исследования, относится к третьей категории. Помещения без повышенной опасности – это помещения, в которых отсутствует сырость, высокая температура, токопроводящие полы, токопроводящая пыль, химическая среда. В данную категорию входят помещения, характеризующиеся пониженной влажностью воздуха (до 75%), оборудованные при необходимости вентиляционной системой и отоплением. Кроме того, полы в таких помещениях должны быть не токопроводящими.

7.4.2 Освещенность рабочего места

Требования к освещению на рабочих местах, оборудованных ЭВМ:

1) Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

2) Искусственное освещение в помещениях для эксплуатации ЭВМ должно осуществляться системой общего равномерного освещения.

3) В производственных и административно-общественных помещениях, в условиях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

4) Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк.

5) Освещение не должно создавать бликов на поверхности экрана.

6) Освещенность поверхности экрана не должна быть более 300 лк. [31]

7) В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ).

Таблица 12 – Нормируемые показатели освещенности для работы с ЭВМ

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность, %	Искусственное освещение		Естественное освещение	
			Освещённость на рабочей поверхности от системы общего освещения, лк	Коэффициент пульсации освещенности КП, %, не более	КЕО, %, при	
					верхнем или комбинированном	боковом
Средней точности	От 0,5 до 0,1	Не менее 70	200	5	4	1,5
		Менее 70	150	10	4	1,5

Расчет необходимого освещения производится методом светового потока:

$$F = \frac{E \cdot K \cdot S \cdot Z}{n}, \quad (55)$$

где F – рассчитываемый световой поток, Лм; E – нормированная минимальная освещенность, Лк; работа оператора относится к разряду точных работ (E = 300Лк); S – площадь освещаемого помещения (S = 21 м^2); Z – отношение средней освещенности к минимальной (Z = 1,1 согласно СНиП 23-05-95 [40]); K – коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (примем K равное 1,5); n – коэффициент использования (определяется по таблице коэффициентов использования различных светильников).

Для нахождения коэффициента n необходимо определить индекс помещения I:

$$I = \frac{S}{h \cdot (a + b)}, \quad (56)$$

где S – площадь помещения, S = 21 м^2; h – расчетная высота подвеса h= 3,8 м; a – ширина помещения, равная 3 м; B – длина помещения, равная 7 м. Тогда показатель I равен:

$$I = \frac{21}{3{,}8 \cdot (3 + 7)} = 0{,}55$$

Поскольку кабинет разработчика имеет свежепобеленный потолок и свежопобеленные стены без штор, то согласно источнику [41], их коэффициенты отражения равны 70% и 50% соответственно. Тогда при использовании люминесцентных ламп серии ОДР и I = 0,28 значение n = 0,28.

Тогда F равно:

$$F = \frac{300 \cdot 1{,}5 \cdot 21 \cdot 1{,}1}{0{,}28} = \ 37125 \text{ Лм}$$

Остается рассчитать необходимое количество ламп, которое находится по формуле:

$$N = \frac{F}{F_л}, \quad (57)$$

где N – определяемое число ламп; F – световой поток; $F_л$ – световой поток лампы (для ламп люминесцентных серии ОДР $F_л$ = 3350 Лм)

Тогда N составляет:

$$N = \frac{37125}{3350} = 11$$

По результатам расчетов кабинет должен быть оборудован 11 светодиодными лампами.

Кабинет разработчика оборудован 12 аналогичными лампами (2 ряда по 6 ламп), следовательно можно говорить о достаточности освещения в кабинете.

7.4.3 Повышенный уровень шума

Характеристикой постоянного шума на рабочих местах являются уровни звукового давления в дБ в октавных полосах со среднегеометрическими частотами 31,5; 63; 125; 250; 500; 1000; 2000; 4000; 8000 Гц, определяемые по формуле:

$$\mathrm{L} = 20 * lg \frac{P}{P_0} \quad (58)$$

где P - среднеквадратичная величина звукового давление, Па; P_0 – исходное значение звукового давления в воздухе, равное 2*10^(-5) Па.

Для различных категорий рабочих помещений нормативные уровни шума регламентируются ГОСТ 12.1.003-2014 ССБТ [34]. Помещения для работы с ПЭВМ не могут граничить с помещениями, в которых присутствует повышенный уровень шума. При выполнении работы на ПЭВМ уровень шума на рабочем месте не должен превышать 50 дБа.

Главными источниками шума в кабинете разработчика являются речь людей и работа офисного оборудования (принтера), при этом уровень шума, создаваемый обычной речью человека и работой принтера, не превышает 60

дБа. Поэтому можно говорить о том, что в целом за рабочий день уровень шума находится в пределах нормы.

7.4.4 Микроклимат в помещении

В помещениях жилых и общественных зданий следует обеспечивать оптимальные или допустимые параметры микроклимата в обслуживаемой зоне. Микроклимат производственных помещений – это климат внутренней среды этих помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха Оптимальные величины показателей микроклимата необходимо соблюдать на рабочих местах производственных помещений, на которых выполняются работы, связанные с нервно-эмоциональным напряжением (на постах управления технологическими процессами, в залах вычислительной техники и др.). Согласно нормативно-технической документации при нормировании параметров микроклимата выделяют холодный период года, характеризуемый среднесуточной температурой наружного воздуха, равной +10°С и ниже и теплый период года, характеризуемый среднесуточной температурой наружного воздуха выше +10°С. Разграничение работ по категориям осуществляется на основе интенсивности общих энергозатрат организма в ккал/ч (Вт). [33]

Офисный кабинет является помещением Iа категории (с интенсивностью энергозатрат до 120 ккал/ч, производимые сидя и сопровождающиеся незначительным физическим напряжением), поэтому должны соблюдаться следующие требования, приведенные в таблице 13.

Таблица 13 – Оптимальные параметры микроклимата

Период года	Категория работ по уровню энергозатрат, Вт	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia (до 139)	(22-24)	(21-25)	(60-40)	0,1
Теплый	Ia (до 139)	(23-25)	(22-26)	(60-40)	0,1

Допустимые величины интенсивности теплового облучения работающих на рабочих местах от производственных источников, нагретых до темного свечения (материалов, изделий и др.) должны соответствовать значениям, приведенным в таблице 14 [38].

Таблица 14 – Допустимые величины интенсивности теплового облучения

Облучаемая поверхность тела, %	Интенсивность теплового облучения, Вт/м2, не более
50 и более	35
25-50	70
не более 25	100

В помещениях, оборудованных ПЭВМ, проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы на ЭВМ.

Для создания и автоматического поддержания в лаборатории независимо от наружных условий оптимальных значений температуры, влажности, чистоты и скорости движения воздуха, в холодное время года используется водяное отопление, в теплое время года применяется кондиционирование воздуха. Кондиционер представляет собой вентиляционную установку, которая с помощью приборов автоматического регулирования поддерживает в помещении заданные параметры воздушной среды.

7.5 Электробезопасность

Источниками электрической опасности являются электрические сети, ПЭВМ и периферийные устройства. При работе с ПК возможен удар током при соприкосновении с токоведущими частями оборудования.

В соответствии с СанПиН 2.2.2/2.4.1340-03 помещения, где размещаются рабочие места с ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации электроустановок и вычислительной техники. [31]

Электрические изделия по способу защиты человека от поражения электрическим током подразделяются на пять классов: 0, 01, 1, 2, 3 [32].

ЭВМ относится к классу 01, то есть, к изделиям, имеющим рабочую изоляцию, элемент для заземления и провод без заземляющей жилы для присоединения к источнику питания.

Для предотвращения возникновения опасных ситуаций обязательны следующие меры предосторожности:

1) Перед началом рабочей смены необходимо убедиться, что выключатели и розетки закреплены и не имеют оголенных токоведущих частей;

2) При обнаружении неисправности оборудования и приборов, необходимо сообщить ответственному лицу, не делая никаких самостоятельных исправлений;

3) Запрещено загромождать рабочее место лишними предметами.

7.6 Экологическая безопасность

В данном подразделе рассматривается характер воздействия проектируемого решения на окружающую среду. Выявляются предполагаемые источники загрязнения окружающей среды, возникающие в результате реализации предлагаемых в работе решений.

7.6.1 Анализ влияния объекта на окружающую среду

Программный модуль – является информационным продуктом и не наносит вреда окружающей среде. С точки зрения влияния на окружающую среду можно рассмотреть влияние ПК и серверного оборудования при их утилизации. Большинство компьютерной техники содержит бериллий, кадмий, мышьяк, поливинилхлорид, ртуть, свинец, фталаты, огнезащитные составы на основе брома и редкоземельные минералы. Это вредные вещества, которые не должны попадать на свалку после истечения срока использования, а должны правильно утилизироваться. Утилизация компьютерного оборудования осуществляется по специально разработанной схеме, которая соблюдается в организации:

1) На первом этапе необходимо создать комиссию, задача которой заключается в принятии решений по списанию морально устаревшей или не рабочей техники, каждый образец рассматривается с технической точки зрения.

2) Разрабатывается приказ о списании устройств. Для проведения экспертизы привлекается квалифицированное стороннее лицо или организация.

3) Составляется акт утилизации, основанного на результатах технического анализа, который подтверждает негодность оборудования для дальнейшего применения.

4) Формируется приказ на утилизацию. Все сопутствующие расходы должны отображаться в бухгалтерии.

5) Утилизацию оргтехники обязательно должна осуществлять специализированная фирма.

6) Получается специальная официальной формы, которая подтвердит успешность уничтожения электронного мусора.

После оформления всех необходимых документов, компьютерная техника вывозится со склада на перерабатывающую фабрику. Все полученные

в ходе переработки материалы вторично используются в различных производственных процессах.

7.7 Безопасность в чрезвычайных ситуациях

7.7.1 Анализ вероятных ЧС, которые может инициировать объект исследований и обоснование мероприятий по предотвращению ЧС

Согласно ГОСТ Р 22.0.02-94 ЧС – это нарушение нормальных условий жизни и деятельности людей на объекте или определенной территории (акватории), вызванное аварией, катастрофой, стихийным или экологическим бедствием, эпидемией, эпизоотией (болезнь животных), эпифитотией (поражение растений), применением возможным противником современных средств поражения и приведшее или могущее привести к людским или материальным потерям".

С точки зрения выполнения проекта характерны следующие виды ЧС:

1) Пожары, взрывы;
2) Внезапное обрушение зданий, сооружений;
3) Геофизические опасные явления (землетрясения);
4) Метеорологические и агрометеорологические опасные явления.

Так как объект исследований представляет из себя программный модуль, работающий на сервере, то наиболее вероятной ЧС в данном случае является пожар в серверной. В серверной комнате применяется дорогостоящее IT-оборудование, не горючие и не выделяющие дым кабели. Таким образом возникновение пожаров происходит из-за человеческого фактора, в частности, это несоблюдение правил пожарной безопасности. К примеру, замыкание электропроводки – в большинстве случаев тоже человеческий фактор. Соблюдение современных норм пожарной безопасности позволяет исключить возникновение пожара в серверной комнате.

- Согласно СП 5.13130.2009 предел огнестойкости серверной должен быть следующим: перегородки - не менее EI 45, стены и перекрытия - не менее

REI 45. Т.е. в условиях пожара помещение должно оставаться герметичным в течение 45 минут, препятствуя дальнейшему распространению огня.

- Помещение серверной должно быть отдельным помещением, функционально не совмещенным с другими помещениями. К примеру, не допускается в помещении серверной организовывать мини-склад IT-оборудования или канцелярских товаров.
- Дверь в помещение серверной также должна обеспечивать требования по огнестойкости (не менее EI 45).
- Необходимо использовать противопожарную дверь.
- При разработке проекта серверной необходимо учесть, что автоматическая установка пожаротушения (АУПТ) должна быть обеспечена электропитанием по первой категории (п. 15.1 СП 5.13130.2009).
- Согласно СП 5.13130.2009 в системах воздуховодов общеобменной вентиляции, воздушного отопления и кондиционирования воздуха защищаемых помещений следует предусматривать автоматически закрывающиеся при обнаружении пожара воздушные затворы (заслонки или противопожарные клапаны).

7.7.2 Анализ вероятных ЧС, которые могут возникнуть при проведении исследований и обоснование мероприятий по предотвращению ЧС

При проведении исследований наиболее вероятной ЧС является возникновение пожара в помещении. Пожарная безопасность должна обеспечиваться системами предотвращения пожара и противопожарной защиты, в том числе организационно-техническими мероприятиями [13].

Под пожарной профилактикой понимается обучение пожарной технике безопасности и комплекс мероприятий, направленных на предупреждение пожаров.

Задачи пожарной профилактики можно разделить на следующие комплекса мероприятий:

1) Организационные мероприятия предусматривают:

- противопожарный инструктаж обслуживающего персонала;
- обучение персонала правилам техники безопасности;
- издание инструкций, плакатов, планов эвакуации.

2) Эксплуатационные мероприятия:

- соблюдение эксплуатационных норм оборудования;
- обеспечение свободного подхода к оборудованию.
- содержание в исправности изоляции токоведущих проводников.

Согласно НПБ 104-03 "Проектирование систем оповещения людей о пожаре в зданиях и сооружениях" для оповещения о возникновении пожара в каждом помещении установлены дымовые оптико-электронные автономные пожарные извещатели, а оповещение о пожаре должно осуществляться подачей звуковых и световых сигналов во все помещения с постоянным или временным пребыванием людей.

При обнаружении пожара осуществляется следующий порядок действий:

1. Сообщить в пожарную охрану по телефону 01 или 112;
2. Оповестить лиц, находящихся в здании, о пожаре;
3. Предпринять действия по прекращению пожара;
4. При опасности поражения электрическим током отключить электроэнергию;
5. Эвакуироваться.

Все помещения оснащаются средствами пожаротушения, а именно огнетушителями типа ОУ-2, ОУ-5 или ОП-5 (предназначены для тушения любых материалов, предметов и веществ, применяется для тушения ПК и оргтехники).

Согласно НПБ 105-03 кабинет, в котором осуществлялась разработка ПМ, является помещением, предназначенным для проектирования и использования результатов проекта, относится к типу В1 – пожароопасное:

Таблица 15 – Категории помещений по взрывопожарной и пожарной опасности

Категория помещения	Характеристика веществ и материалов, находящихся (обращающихся) в помещении
В1 пожароопасные	Горючие и трудногорючие жидкости, твердые горючие и трудногорючие вещества и материалы (в том числе пыли и волокна), вещества и материалы, способные при взаимодействии с водой, кислородом воздуха или друг с другом только гореть, при условии, что помещения, в которых они имеются в наличии или обращаются, не относятся к

Вывод по разделу

В ходе написания раздела «социальная ответственность» были выявлены и проанализированы наиболее вероятные вредные и опасные производственные факторы, а также предложены мероприятия по снижению уровней их воздействия на работника. Также рассмотрены наиболее возможные чрезвычайные ситуации на рабочем месте и алгоритм действий при их возникновении.

Анализ влияние факторов показывает, что существенных нарушений по организации работы нет. Поставленные требования и нормы безопасности соблюдены, а организационные вопросы по обеспечению рабочих условий подтверждены законодательно и не нарушают законодательный регламент.

Заключение

В период работы над магистерской диссертацией было изучено одно из наиболее популярных направлений в настоящее время – машинное обучение. Для выполнения целей и задач, поставленных в работе, был изучен основной стек технологий, включающий язык программирования, среду разработки и необходимые библиотеки, обоснование применения которых описано в главе 2.

Глава 3 полностью посвящена методам обработки исходной информации, которые могут быть применены практически к любому набору данных, используемому в дальнейшем для реализации методов машинного обучения.

Также в работе (глава 4) описаны основные методы машинного обучения, приведено математическое описание и обоснование используемых в них алгоритмов. Были рассмотрены как классические методы, так и наиболее современные разработки. Оценка точности моделей осуществлялась по различным метрикам качества для более комплексного анализа. По результатам оценок лучшим методом оказался “XGBoost” с показателями точности 77% для детектирования и 88% для классификации дефектов. Показатели точности не позволяет полностью доверяться вышеупомянутой модели на данный момент, однако она может использоваться как инструмент для первичного анализа качества. Дальнейшее накопление данных позволит только увеличить точность модели, в результате чего она сможет играть роль экспертной системы. Интеграция подобной модели в 3-х уровневую АСУТП представляет собой 4-ый уровень (уровень управления цехом), на котором будут решаться задачи планирования ресурсов, управления техническим обслуживанием и другие.

Используемые в работе методы и модели являются гибким инструментом, которые могут применяться для контроля качества практически в любой

отрасли, в связи с чем их применение особенно актуально в условиях нацеленности большинства предприятий на цифровизацию производства.

Список литературы

1. Денисов И.В., Смирнов А.А. методика проведения входного контроля качества // Современные проблемы науки и образования. – 2013. – № 5.
2. И.П. Мазур, “Контроль качества поверхности листового проката, ФГБОУ ВПО «Липецкий государственный технический университет» - URL: http://elar.urfu.ru/bitstream/10995/33298/1/itvmim_2012_63.pdf
3. Новокщенова С.М., Виноград М.И. Дефекты стали. – М.: Металлургия, 1984. – 199с.
4. Методы получения заготовок деталей машин: учебное пособие / В. Ф. Пегашкин, Е. В. Пегашкина; М-во образования и науки РФ; ФГАОУ ВПО «УрФУ им. первого Президента России Б.Н.Ельцина», Нижнетагил. техн. ин-т (филиал). − Нижний Тагил: НТИ (филиал) УрФУ, 2016. – 81 с.
5. Непрерывная разливка стали и сплавов: учебное пособие / Н.А. Козырев, Р.А. Гизатулин, Д.В. Валуев; Юргинский технологический институт. – Томск: Изд-во Томского политехнического университета, 2014. – 406с.
6. Системы контроля качества поверхности на линиях производства –URL: http://www.tadviser.ru/index.php/Статья:Как_системы_компьютерного_зрения_помогают_контролировать_качество_продукции
7. Высокопроизводительный контроль качества поверхности - Алексей В. Белобородов, Евгений В. Власов, Петр С. Завьялов, Леонид В. Финогенов, 2015. – 147 с.
8. Машинное обучение – подготовка данных, – URL: https://coderlessons.com/tutorials/python-technologies/uznaite-mashinnoe-obuchenie-s-python/mashinnoe-obuchenie-podgotovka-dannykh
9. Подготовка данных для алгоритмов машинного обучения, – URL: http://blog.datalytica.ru/2018/04/blog-post.html

10. Введение в машинное обучение с помощью Python, Мюллер А., Гвидо С., 2017.

11. Библиотека NumPy, начало работы – URL: https://pythonworld.ru/numpy/1.html

12. Документация библиотеки Seaborn – URL: https://seaborn.pydata.org/

13. Документация метода XGBoost – URL: https://xgboost.readthedocs.io/en/latest/parameter.html

14. Документация библиотеки Pandas – URL: https://pandas.pydata.org/

15. Документация Jupyter Notebook – URL: https://jupyter.org/

16. Python IDEs and Code Editors (Guide) – URL: https://realpython.com/python-ides-code-editors-guide/

17. М.А. Харченко, Корреляционный анализ, Учебное пособие для вузов, 2015. – 32 с.

18. Что такое машинное обучение? Методы, типы, задачи и примеры машинного обучения, – URL: https://mining--cryptocurrency-ru.turbopages.org/s/mining-cryptocurrency.ru/mashinnoe-obuchenie-metody-tipy/

19. Введение в машинное обучение, – URL: https://ru.linkedin.com/in/grigorysapunov

20. Машинное обучение. Разинков Е.В., КСАИТ, ИВМиИТ КФУ – URL:https://vmkhelp.ru/wp-content/uploads/2017/09/machine_learning_at_kfu_25_04_15.pdf

21. Решающее дерево (Decision tree) – URL: https://learnmachinelearning.wikia.org/ru/wiki/Решающее_дерево_(Decision_tree)

22. Руководство к использованию деревьев решений в машинном обучении и науке о данных – URL: https://medium.com/nuances-of-programming/руководство-к-использованию-деревьев-решений-в-машинном-обучении-и-науке-о-данных-c10030f05349

23. Обзор методов классификации в машинном обучении с помощью Scikit-Learn – URL: https://tproger.ru/translations/scikit-learn-in-python/

24. Чусовлянов Д.С. Машинное обучение для определения тональности и классификации текстов на несколько классов. Выпускная квалификационная работа бакалавра. – НИУ ВШЭ. – 2014

25. Логистическая регрессия – URL: https://conf.sfu-kras.ru/sites/mn2012/thesis/s021/s021-083.pdf

26. К. В. Воронцов. Математические методы обучения по прецедентам (теория обучения машин). Курс лекций. – URL: http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf

27. Бурков Андрей, Б91 Машинное обучение без лишних слов. — СПб.: Питер, 2020. — 192 с.

28. ГОСТ Р ИСО 16269-4-2017 Статистические методы. Статистическое представление данных. Часть 4. Выявление и обработка выбросов

29. Библиотека NumPy, начало работы – URL: https://scikit-learn.org/stable/#

30. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем.Пер. с англ. - СпБ.: ООО "Альфа-книга': 2018. - 688 с

31. Визуальный и измерительный контроль: учебное пособие для подготовки специалистов 1, 2 и 3 уровня / Н.П. Калиниченко, А.Н. Калиниченко; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2009. – 300 с.

32. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы

33. СанПиН 2.2.4.1191-03 Электромагнитные поля в производственных условиях

34. СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений

35. ГОСТ 12.1.003-2014 Система стандартов безопасности труда (ССБТ). Шум. Общие требования безопасности (Переиздание)

36. ТК РФ Статья 108. Перерывы для отдыха и питания

37. ГОСТ 12.2.032-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования

38. ГОСТ 12.0.003-2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация

39. СанПиН 2.2.4.3359-16 Санитарно-эпидемиологические требования к физическим факторам на рабочих местах

40. Правила устройства электроустановок. Седьмое издание, 2002

41. СНиП 23-05-95* Естественное и искусственное освещение

42. Бородин Ю.В., Василевский М.В., Дашковский А.Г., Назаренко О.Б., Свиридов Ю.Ф., Чулков Н.А, Федорчук Ю.М. Д 12 Безопасность жизнедеятельности. Практикум: учебное пособие по выполнению индивидуальных заданий для студентов всех специальностей – Томск: Издательство Томского политехнического университета, 2009. – 50 с;

43. ГОСТ 12.2.007.0-75 ССБТ. Изделия электротехнические. Общие требования безопасности (с Изменениями N 1, 2, 3, 4);

44. ГОСТ 12.1.004-91 ССБТ. Пожарная безопасность. Общие требования;

45. Документация библиотеки Snap7 – URL: https://python-snap7.readthedocs.io/en/latest/

Приложение А (справочное) Раздел на английском языке

(справочное)

Detection and classification of defects in metal workpieces using machine learning methods

Студент

Группа	ФИО	Подпись	Дата
8ТМ81	Маляров Дмитрий Владимирович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Евгений Иванович	к.т.н.		

Консультант-лингвист отделения иностранных языков ШБИП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ОИЯ ШБИП	Пичугова Инна Леонидовна	–		

Description of the process

To determine the relationship between the source dataset and the process, as well as to understand the composition and parameters of data in the dataset, it is necessary to understand the basics of the casting process, the causes and types of possible defects, and etc.

The process begins with the delivery of molten steel in a steel bucket to the continuous casting machine and placing it in the casting position.

Continuous casting of steel is carried out on continuous casting machines (CCM) and can be used in all steelmaking industries. Continuous casting is most widely used in converter shops.

According to the design of the CMM for casting steel they are divided into vertical, radial and curved (horizontal CMM are being developed). A more modern design has radial and curved CMM. A feature of such machines is a bend with a certain radius of the mold, which forms a correspondingly curved ingot.

After leaving the mold, the ingot enters a rigid guide channel of secondary cooling, consisting of roller sections, and passes through the process of crystallization of 1/4 circle. The radius of the circle is chosen so that the ingot does not contain a liquid phase when moving to the horizontal position. A special feature of curved CMM is the bending of the ingot with a variable radius. After moving to the horizontal position, the continuously cast ingot is straightened in the correct-pulling crates and cut into dimensional blanks.

Continuous casting machines are complex multi-machine units with a large number of automated electric drives, units and systems for automatic control and regulation. The diagram of the vertical CMM and the automation system is shown in Figure 1. Steel is fed from the steelmaking compartment in the bucket, from which it is poured into the intermediate bucket and then into the mold. The ingot with hardened walls is pulled down by a pull cage, passing through the secondary cooling zone with water. The ingot is cut into measured lengths by automatic gas cutting.

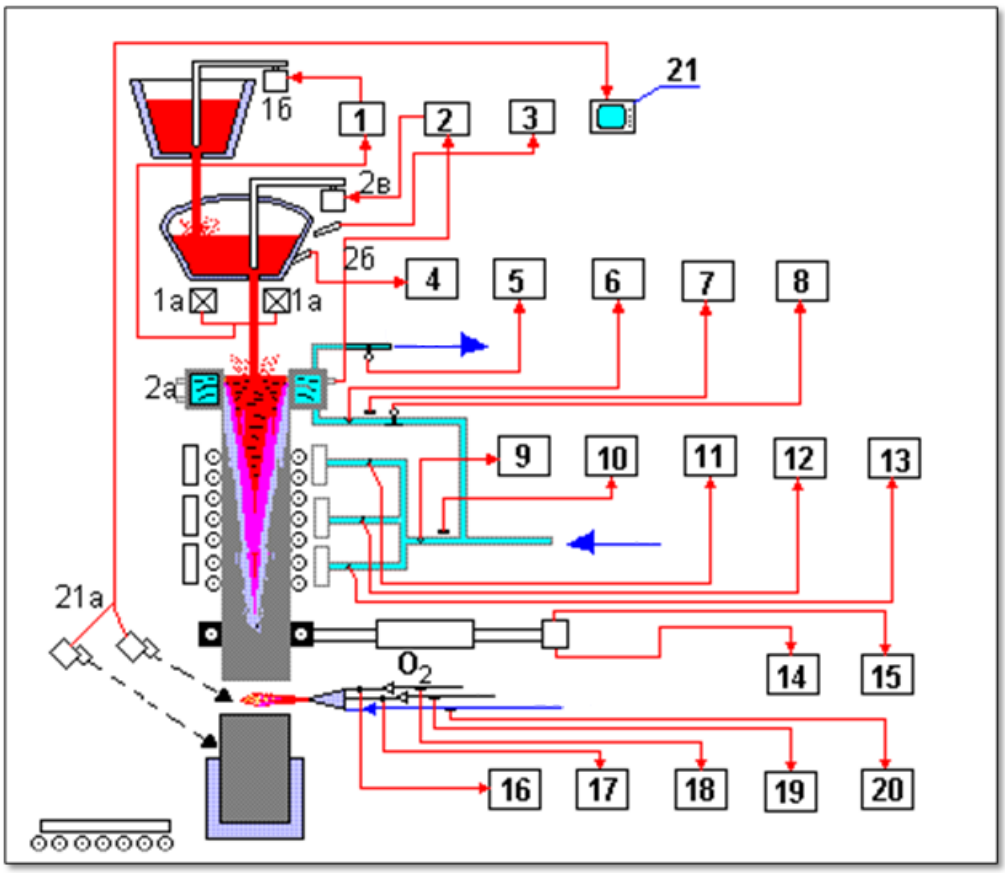


Figure 1. The automation of the continuous casting machine of vertical type

Automated electric drives operate the pulling crate, the mechanism for swinging the mold, the mechanisms for cutting gas, issuing ingots, the correct machine, etc. For the convenience of monitoring and controlling the CMM mechanisms, the automation panel provides a mnemonic circuit with an alarm about the state of the mechanisms and an alarm panel. Emergency and warning alarms indicate that the mold swing mechanism is disabled, the pulling cage is stopped, the ingot is being cut too long, there is no lift truck under the ingot, and so on.

Continuous steel casting plants operate in a stationary mode and require a perfect system of automatic control and regulation to maintain this mode. Deviations from the best mode of casting, caused by a variety of disturbances can lead to decreased performance, deterioration in the quality of the metal and the occurrence of accidents. Automatic control and regulation systems of the CMM contribute to the elimination of disturbances and ensure the most rational mode of casting and safe operation of the unit.

The main functions of the system of control and regulation of the casting process are:

a) control and automatic stabilization of the liquid metal levels in the intermediate bucket and the crystallizer, which ensures a uniform discharge of the metal and a stationary mode of its crystallization, necessary to obtain a good quality ingot;

b) control and regulation of water flow in the sections of the secondary cooling zone for uniform heat removal from the ingot, which is also necessary to obtain a good quality of the metal;

c) monitoring of the temperature state of the structural elements of the unit in order to eliminate emergency modes;

d) automatic cutting of the ingot into measured lengths, i.e. into blanks of a given length.

The metal level in the intermediate bucket (Fig. 1) is stabilized by a set of equipment consisting of strain gauges of mass 1a and a regulating device 1 that controls the drive of the bucket stopper 1b. Thus, the level of metal in the intermediate bucket is regulated indirectly by stabilizing its mass. The control device operates according to a two-position control law.

Stabilization of the metal level in the mold provides stationary conditions for solidification of the ingot and its good quality, as well as safe operation of the installation. Exceeding the level of metal in the mold can lead to the overflow of steel to the work site, and an unacceptable decrease to the breakout of liquid metal from the internal volume of the ingot through an insufficiently formed crust after leaving the mold. Both cases are emergency.

The equipment of the level control loop consists of a radioactive level transmitter having a source of gamma radiation 2a and a receiver 2b, a control set 2 and a stop drive of the intermediate bucket 2c. When the metal level deviates from the set value, the regulator lowers or raises the stopper, thereby reducing or increasing the flow section of the drain hole in the bottom of the bucket.

The temperature of the steel in the intermediate bucket is periodically monitored by the immersion thermocouple with registration on the potentiometer 3 to monitor the degree of heating of the bucket masonry before filling with metal, a thermocouple with a recording potentiometer 4 is installed in it.

When casting ingots of a small cross-section, the overlap of the metal jet with a stopper can lead to its deformation, metal splashing along the walls of the mold and deterioration of the quality of the ingot formation. Therefore, a method is used to regulate the level of metal in the mold by changing the speed of drawing the ingot with a constant supply of liquid metal from the intermediate bucket. In this case, the controller 2 acts on the drive of the pulling cage.

Regulation of water flow in the sections of the secondary cooling system is necessary to organize the correct mode of crystallization and cooling of the metal along the height of the ingot and along its perimeter. Uniform cooling of the ingot faces eliminates possible deformation due to temperature stresses. Water consumption in the secondary cooling sections is controlled by standard sets 11, 12, 13 with measuring diaphragms or rotameters as primary devices. Changing the water flow rate is carried out by remote manual control of the control valves on the water pipes.

The pressure and flow of water to the mold and secondary cooling are controlled by devices 6, 7 and 9, 10, and the pressure gauges 7 and 10 are equipped with signal contacts to signal an unacceptable drop in water pressure.

Control of the thermal operation and temperature state of the mold is carried out by measuring the water temperature at its outlet with a resistance thermometer with an electronic automatic bridge 5. A similar set 8 controls the water temperature at the entrance to the mold. The total length of the ingot and measured lengths are counted using pulse sensors mounted on the shaft of the pulling cage reducer and the device 14, which includes pulse counters and indicating indicators. Tachometer and the instrument 15 are determined by the speed of movement of the metal.

The work of the automatic machine requires the appropriate amounts of gas, oxygen and cooling water. The pressure in the supply lines is controlled by manometric sets with signal contacts 18, 19, 20, and gas and oxygen consumption-measuring diaphragms with devices 16 and 17.

To monitor the operation of individual parts of the unit, for example, the operation of the gas cutting machine and the mechanism for receiving and issuing cut ingots, an industrial television installation is used, consisting of cameras and an image receiver 27. During research and adjustment, the temperature of the ingot at various sites is controlled using radiation pyrometers.

The operation of the entire CMM is controlled by the central operator panel located on the filling platform, the gas cutting panel and the control panel for issuing ingots. The Central control panel provides remote start and stop of the machine, regulates the speed of pulling the ingot, turns on and off the water cooling, the mechanism for swinging the mold, and the supply of lubricant. If there is no automatic control of metal levels in the intermediate bucket and the mold, the operator remotely controls the bucket stoppers from the central console. For operational communication between the control panels of the CMM and between the installation and other parts of the shop, a loudspeaker is used.

The advantage of installations of this type is that they require a lower height for their construction of the shop due to the bending of the ingot, and the total capital costs for the construction of shops are reduced. The basic functions of management, controls and components of the control system for radial continuous casting machine are the same as for vertical installations.

The level of metal in the intermediate bucket and the mold is regulated by blocks 1 and 2, blocks 3 and 4 are designed to regulate the cooling of the mold and metal, blocks 5 and 6 regulate the flow of gas and oxygen to the gas cutter.

The difference between radial CMMs is also that they do not use locking devices, but gate devices to control the discharge of metal from the filling bucket to the intermediate one and from the latter to the mold.

Local systems include:

– automatic control system of the mixing department of the steelmaking shop;

– automatic control system of level in the intermediate bucket;

– automatic control system of metal level in the mold;

– automatic control system of the thermal mode of the mold;

– automatic control system of secondary cooling, etc.

Automatic control system for continuous casting of steel

A continuous casting process control system is usually part of an integrated steelmaking control system, such as a Converter shop. In general, the automated process control system should provide, by stabilizing and optimizing the technological modes of casting, increased productivity; increased yield of usable metal; reducing the number of emergency modes of operation and improving the efficiency of the CMM, improving the working conditions of service personnel.

The main functions of the automatic control system for continuous casting of steel:

A) Information and information-computing systems;

1. control of values;

a) temperature of liquid steel in the steel ladle;

b) the temperature of the liquid steel in the intermediate bucket;

c) the mass of steel in the steel ladle;

d) the mass (level) of the metal in the intermediate bucket;

e) the level of metal in the mold;

f) efforts to pull the ingot out of the mold;

g) the rate of withdrawal of the ingot (casting speed);

h) cooling water flow rate and pressure on the mold;

i) temperature difference of cooling water on the mold;

j) the flow rate of technological lubricant in the mold;

k) water flow and pressure on the secondary cooling zone sections;

l) ingot surface temperature;

m) forces on the support rolls of the ingot correction section;

n) the total and measured lengths of the ingot.

2. Calculation;

a) the thermal state and thickness of the ingot shell in the secondary cooling zone;

b) the main parameters of casting (speed, lubricant consumption, cooling water consumption for the mold and for secondary cooling);

c) technical and economic indicators of the operation.

Control function:

1. managing values;

a) the mass (level) of the metal in the intermediate bucket:

b) the level of metal in the mold;

c) water flow to the mold;

d) water consumption in the secondary cooling zone sections;

e) flow rate of technological lubricant:

f) gas and oxygen consumption for the gas cutting machine.

2. Process control;

a) starting mode continuous-casting machine;

b) secondary cooling mode of the ingot;

c) ingot hoods (driven by pulling crates;

d) cut the ingot into measured lengths;

e) the optimal mode for the end of casting in order to reduce waste;

f) the "melt-to-melt" method of casting by calculating and issuing recommendations for maintaining the desired contact schedule.

Appropriate monitoring and control systems are provided for the implementation of optimal functions.

In addition to the above list of functions, the automated process control system performs:

– signaling deviations from the norms of the main technological parameters of the process;

– accumulation of information about the casting mode and conditions of formation of each billet for subsequent analysis;

– registration of pre-emergency situations;

– preparation and printing of the technological passport of casting and other documents about the work of the CMM.

Surface quality control system

Manual quality control still prevails in many types of production, as any industrial enterprise has a whole arsenal of trained employees and honed quality standards that products are checked for compliance with.

Currently, the automated visual quality control system can significantly reduce the direct participation of employees in the quality control process on all types of production lines, giving a person the role of the process manager [6].

A system for detecting defects in steel billets, rolling sheet and etc. on the conveyor. may consist of several cameras, including infrared cameras, structured light illumination systems, and data storage and processing servers with workstations.

Thanks to the SQCS, various parameters of defects in the workpieces are removed, such as coordinates, the number of pixels per defect area, etc.

Parameter values are recorded on the data recording server and then transmitted to the data center, where defects are classified based on the information received. After classification, the processing results are transmitted to the control station, informing the mill operator about the surface condition, and also recorded in the database for further analysis of the quality of rolled products.

Tools for working with data and machine learning methods

Selecting a programming language

You can analyze data and create models for using machine learning methods in several programming languages:

1) Python;

Python is a high-level programming language that has many different uses, including data science and internal web development. It is a powerful tool for data analysis, widely used in big data technology.

Thanks to the active Python community, there are many ready-made machine learning libraries.

This language is platform-independent, so it can be adapted to almost any operating system.

One drawback is the difficulty of tracking errors in the code.

2) R;

R is widely used in data analysis and is usually targeted for solving general machine learning problems, such as regression, classification, and decision tree formation.

Like Python, R is open source and widely known as a language that is relatively easy to install, configure, and apply.

R is also platform-independent and integrates well with other programming languages. Along with data analysis, R is adapted for data visualization.

Despite the relative ease of integration with other tools, R has a number of features that make it difficult to learn. These include, for example, non-traditional data structures and indexing (which starts with 1 instead of 0).

R is less popular than Python, so it has less documentation required by developers to create applications in the machine learning area.

3) JavaScript;

This language appeared in the mid-1990s as a tool for improving the practice of web development and is one of the most popular in this field.

As for the advantages of JavaScript in the field of machine learning, it opens up opportunities to enter an uncharted path easier for web and app developers who are largely already familiar with it. However, the current JavaScript ecosystem for

machine learning still looks immature, so support for this type of development is currently limited.

In addition, the language lacks functions for working with data, which are present by default in languages such as R and Python.

4) C++.

C++ is the oldest of the most widely used programming languages today. With the capabilities of both a low-level and high-level programming language, C++ provides a higher level of control and efficiency than other programming languages in the context of machine learning.

The flexibility of the language is well suited for resource – intensive applications, and the subset of machine learning programs is no exception. Given that C++ is a statically typed language, it can perform tasks at a relatively high speed.

As for its disadvantages, the main one is that creating new applications based on C++ requires writing a large amount of complex code, which takes a lot of time and can cause great difficulties in maintenance, as a result, the C++ language is difficult to master

Analyzing the advantages and disadvantages of the presented languages, as well as taking into account the current level of proficiency in them, Python was chosen as the main language for data processing and building models.

Development environment

When writing code in Python, integrating modules and libraries to build large systems, a text editor is not enough. An integrated development environment (IDE) is required.

An IDE is a program designed for software development. As the name suggests, the IDE combines several tools specifically designed for development. These tools usually include an editor designed to work with code (such as syntax highlighting and autocomplete); build, run, and debug tools; and a form of version control [17].

However, interacting with data involves working simultaneously with code, images, graphs, and so on. One of the best solutions is Jupyter Notebook.

Jupyter Notebook is a powerful tool for developing and presenting Data Science, Machine Learning, and other projects in an interactive way. It combines code and output all in a single document containing text, mathematical equations, and visualizations.

This step-by-step approach ensures a fast, consistent development process, since the output for each block is shown immediately.

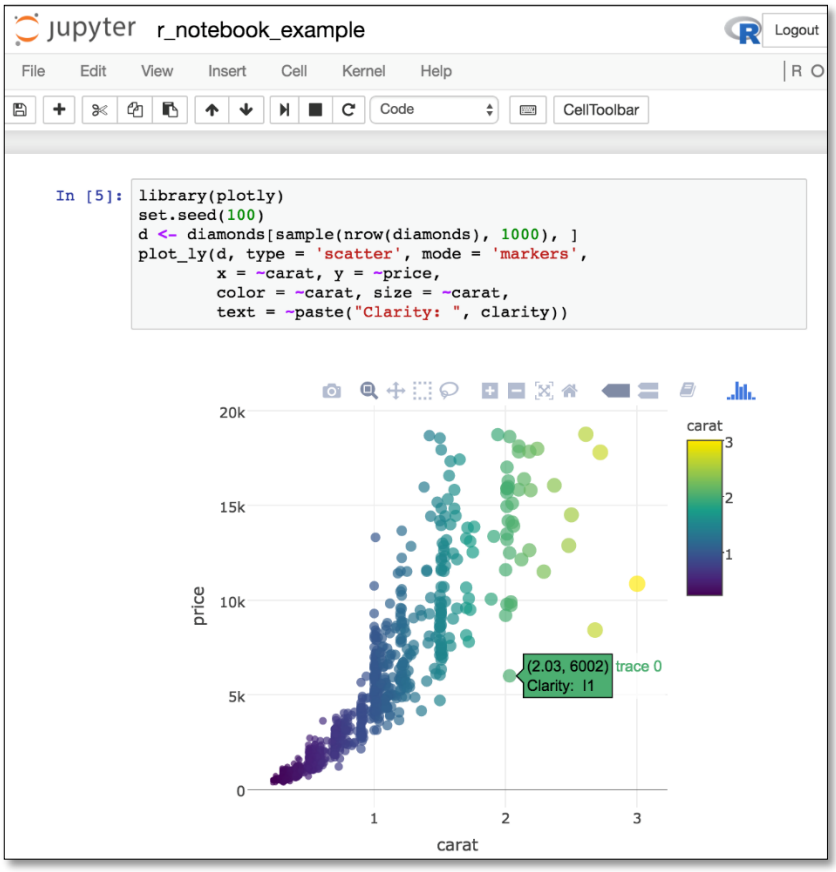


Figure 2. Jupyter Notebook Window

Libraries

In addition to using the basic functions implemented in Python, you also need to connect a number of libraries, which are a set of modules and functions that facilitate a large number of specific operations using this programming language.

These libraries include:

1) Scikit-learn;

This is one of the most popular machine learning libraries. It supports many controlled and unsupervised learning algorithms. For example, linear and logistic regressions, decision trees, clustering, k-means, etc. [29]

It is based on the two main Python libraries – NumPy and SciPy. Scikit-learn adds a set of algorithms for common machine learning and data mining tasks, including clustering, regression, and classification.

2) Pandas;

A library that transforms high-level data structures (such as datasets) into easy-to-use and intuitive ones.

It contains built-in methods for grouping, combining and filtering data, as well as time series analysis [14].

Pandas allows you to extract data from various sources, such as SQL databases, CSV, Excel, and JSON files, and manipulate this data to perform operations with them.

3) Seaborn;

This library is designed for creating statistical graphics in Python. It is built on top of matplotlib and is closely integrated with Pandas data structures [12].

4) NumPy;

NumPy provides general mathematical and numerical operations, including basic methods for manipulating large arrays and matrices [11].

5) SciPy.

Scientific Python extends the NumPy functionality with a huge collection of useful algorithms, such as minimization, Fourier transform, regression, and other applied mathematical techniques [13].

Preparing source data

The initial data for training is the parameters of defects and workpieces measured using the system described in chapter 1.2.

The data is a table where the columns are the names of defects or their classes (types), and the rows are parameter values for a single dimension (one blank).

The parameters of defects include:

• X_Minimum;

• X_Perimeter;

• Y_Perimeter;

• Sum_of_Luminosity;

• Minimum_of_Luminosity;

• SigmoidOfAreas and etc.

Thus, there are 27 features that are input information for training and testing models. Formally, a feature is called the mapping $f : X \to D_f$, where D_f is the set of acceptable values of the feature [26].

Depending on the nature of the set D_f attributes are divided into several types:

A) $D_f = \{0, 1\}$, then f is a binary attribute;

B) D_f is a finite set, then f is a nominal feature;

C) D_f is a finite ordered set, then f is an ordinal feature;

D) $D_f = R$, then f is a quantitative attribute.

The type of attribute directly determines the choice of methods for processing a particular data column.

The remaining 7 columns show whether the defect belongs to a particular class. Classes represented in the selection:

• Pastry;

• Z_Scratch;

• K_Scratch;

• Stains;

• Dirtiness;

• Bumps;

• Other_Faults.

The results of assigning each of the blanks to a specific class (this process is commonly called data markup) are the output data set for the models being trained and tested.

Working with a data set begins by reading the file where it is stored. It is a “csv” file where the values of neighboring features are separated from each other by a comma.

Let us look at the data characteristics for each of the features.

```
data.describe()
```

	X_Minimum	X_Maximum	Y_Minimum	Y_Maximum	Pixels_Areas	Y_Perimeter	Sum_of_Luminosity	Maximum_of_Luminosity	Length_of_Conveyer
count	1945.000000	1945.000000	1.945000e+03	1.945000e+03	1945.000000	1944.000000	1.945000e+03	1945.000000	1945.000000
mean	570.444216	617.202571	1.649672e+06	1.649726e+06	1890.681748	82.880144	2.059616e+05	130.204627	1459.608740
std	520.447843	497.473317	1.774542e+06	1.774554e+06	5163.631152	426.159416	5.118255e+05	18.675215	144.768744
min	0.000000	4.000000	6.712000e+03	6.724000e+03	2.000000	1.000000	2.500000e+02	37.000000	1227.000000
25%	51.000000	192.000000	4.683170e+05	4.685200e+05	84.000000	13.000000	9.522000e+03	124.000000	1358.000000
50%	434.000000	466.000000	1.199744e+06	1.199753e+06	175.000000	25.000000	1.922600e+04	127.000000	1364.000000
75%	1051.000000	1071.000000	2.183073e+06	2.183084e+06	812.000000	83.000000	8.256500e+04	140.000000	1650.000000
max	1705.000000	1713.000000	1.298766e+07	1.298769e+07	152655.000000	18152.000000	1.159141e+07	253.000000	1794.000000

8 rows × 32 columns

```
data.describe()
```

:eel_A300	...	Orientation_Index	Luminosity_Index	SigmoidOfAreas	Pastry	Z_Scratch	K_Scatch	Stains	Dirtiness	Bumps	Other_Faults
15.000000	...	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000	1945.000000
0.400000	...	0.084002	-0.131297	0.585490	0.082776	0.097686	0.201028	0.037018	0.028278	0.207198	0.346015
0.490024	...	0.501154	0.148656	0.339325	0.275615	0.296966	0.400872	0.188854	0.165808	0.405403	0.475821
0.000000	...	-0.991000	-0.998900	0.119000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	...	-0.333300	-0.195000	0.248200	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	...	0.095200	-0.132800	0.506300	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1.000000	...	0.514300	-0.066600	0.999800	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
1.000000	...	0.991700	0.642100	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figure 3. Main parameters of feature values

Analyzing the data, we can conclude that they are homogeneous, i.e. they belong to the same type – a quantitative attribute (not related to class labels). In addition, for several attributes, the difference in values in the minimum – median and median – maximum pair differs several times (for example, the “Pixel_areas” attribute, where the maximum value of the attribute is 152655 with a median of 175). This may indicate that there are outliers (abnormal or incorrect values) in the data.

Let us move on to preparing the data. Before you start training and testing the model, make sure that the data is complete, i.e. it does not contain incorrect values, etc.

1. Checking the completeness of data;

The source data table must not contain empty values. If there is at least one empty value in a row in any of the attributes, this row should be deleted, because this may negatively affect the results of training and testing, or the model will return an error when processing data.

2. Checking for incorrect values;

It consists in searching for abnormal values – “outliers” that also impair the quality of model training. These include values with an excessively large or small value, negative values (for the parameters length, area, etc.), zero values, and so on.

Emissions can distort and reduce the information contained in the data source or generation procedure. In production, the presence of emissions reduces the effectiveness of production processes, product quality, and product control procedures [29].

Graphical analysis can also be used to make the search and finding of “outliers” easier, in particular, the construction of a scattering diagram. A demonstration of this diagram using the “Pixels_Areas” attribute is shown in Figure 4.

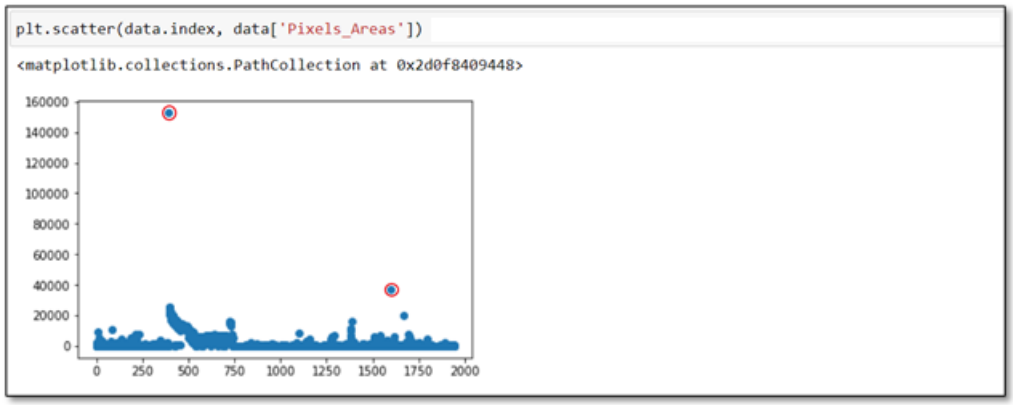


Figure 4. Dot diagram of the “Pixels_Areas”

Similarly, you can apply this method to the other attributes and then delete rows containing outliers from the dataset.

At the moment, the theory of statistical analysis does not have an unambiguous criterion for identifying outliers, so abnormal values can be values that go beyond ±3σ in the case of a normal distribution of data.