

КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
имени АЛЪ-ФАРАБИ

РЕКОМЕНДАЦИИ
ПО ИНТЕГРАЛЬНОЙ ОЦЕНКЕ
РАЗВИТИЯ АПК НА ОСНОВЕ
ПРОСТРАНСТВЕННО-ВРЕМЕННЫХ,
СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ И
ЭКОЛОГИЧЕСКИХ ДАННЫХ

Алматы
«ESKoPrint»
2026

УДК 338.43:332.1

ББК 65.32

Р 36

*Рекомендована к изданию Ученым советом
КазНУ им. аль-Фараби (протокол № 11 от 24.04.2026 г.)*

(выполнено в рамках проекта BR 22886730)

Рецензенты:

Дюсембеков З.Д., доктор сельскохозяйственных наук, профессор
кафедры земельного кадастра, НАО «Казахский агротехнический
исследовательский университет
им. С. Сейфуллина», академик НАН РК.

Бексенова Р.Т., доктор географических наук, профессор кафедры
картографии и геоинформатики, НАО «Казахский национальный
университет им аль-Фараби».

Авторы:

*О.Ә. Әліпбеки, П.П. Гроссул, Ч.А. Аліпбекова,
Д.К. Рахимов, М.М. Аліев, Д. Кенжебек, Қ.К. Аугамбаев*

**Р 36 Рекомендации по интегральной оценке развития АПК на основе
пространственно-временных, социально-экономических и экологи-
ческих данных/коллектив авторов. – Алматы: ESKoPrint, 2026. – 166 с.
ISBN 978-601-04-7643-1**

Рекомендации посвящены разработке и внедрению методических подходов к интегральной оценке развития АПК на примере пространственно-временных, социально-экономических и экологических данных 84 сельских округов Оskarовского, Бухар-Жырауского, Абайского и Каркаралинского районов Карагандинской области. Их применение направлено на преодоление ограниченности традиционных показателей эффективности и обеспечение перехода к многомерному анализу развития АПК. В рекомендации обосновывается необходимость использования больших данных, геоинформационных технологий и алгоритмов машинного и глубокого обучения (искусственного интеллекта) для прогнозирования, выявления скрытых закономерностей и повышения качества аналитического сопровождения аграрной деятельности. Практическая значимость рекомендаций состоит в создании инструментов для подготовки оперативных отчетов, корректировки программ субсидирования, оценки рисков и распространения лучших практик, а также позволит повысить прозрачность управленческих решений, основанных на аграрной инфраструктуре пространственных данных 2.0 и будет способствовать достижению ЦУР-2030.

Рекомендации разработаны по результатам исследований по научно-технической программе «Фор-мирование аграрной инфраструктуры пространственных данных Республики Казахстан по технологиям и принципам 2.0», ИРН: BR 22886730 по линии программно-целевого финансирования Министерства сельского хозяйства Республики Казахстан.

УДК 338.43:332.1

ББК 65.32

ISBN 978-601-04-7643-1

© Коллектив авторов, 2026

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АгроИПД – Аграрная инфраструктура пространственных данных

АПК – агропромышленный комплекс

АТЕ – административно-территориальные единицы

БПЛА – беспилотный летательный аппарат

Веб – всемирная паутина

ВОА – веб-ориентированная архитектура

ГИС – геоинформационные системы

ДЗЗ – дистанционное зондирование Земли

ЗРК – Закон Республики Казахстан

ИК – инфракрасный (диапазон)

КРС – крупный рогатый скот

ЕС – Европейский союз

НИИПД – национальная инфраструктура пространственных данных

ООН – Организация Объединённых Наций

ПД – пространственные данные

ПВД – пространственно-временные данные

СТ – сельские территории

ЦУР - Цели устойчивого развития

ALE – Accumulated Local Effects (накопленные локальные эффекты)

API – Application Programming Interface (программный интерфейс приложения)

BCC – Banker–Charnes–Cooper (модель DEA с переменной отдачей от масштаба)

Big Data (5V) – большие данные: объем (Volume), разнообразие (Variety), скорость (Velocity), достоверность (Veracity), ценность (Value)

BiLSTM – Bidirectional Long Short-Term Memory (двунаправленная рекуррентная нейросеть)

CCR – Charnes–Cooper–Rhodes (модель DEA с постоянной отдачей от масштаба)

CRM - Customer Relationship Management (Управление взаимоотношениями с клиентами)

CRS – Constant Returns to Scale (постоянная отдача от масштаба)
CV – Cross-Validation (кросс-валидация)
DEA – Data Envelopment Analysis (непараметрический метод оценки эффективности)
DL – Deep Learning (глубокое обучение)
DMU – Decision Making Unit (единица принятия решений)
ETL – Extract, Transform, Load (извлечение, преобразование и загрузка данных)
EVI – Enhanced Vegetation Index (улучшенный вегетационный индекс)
FAO – Food and Agriculture Organization (Продовольственная и сельскохозяйственная организация ООН)
FAOSTAT – статистическая база данных FAO
FaST - Farm Sustainability Tools (Инструменты устойчивого развития сельского хозяйства)
FDR – False Discovery Rate (доля ложных открытий)
GEO – Group on Earth Observations (Группа по наблюдениям Земли)
GEOGLAM – GEO Global Agricultural Monitoring (глобальный сельскохозяйственный мониторинг GEO)
GGIM – Global Geospatial Information Management (глобальное управление геопространственной информацией, комитет экспертов ООН)
GeoJSON - JavaScript Object Notation (объектная нотация JavaScript)
GML - Geography Markup Language (язык географической разметки)
GPS – Global Positioning System (глобальная система позиционирования)
ICE – Individual Conditional Expectation (индивидуальные условные ожидания)
IG – Integrated Gradients (интегрированные градиенты)
IGIF – Integrated Geospatial Information Framework (интегрированная структура геопространственной информации)
INSPIRE – Infrastructure for Spatial Information in Europe (инфраструктура пространственной информации Европейского союза)

ISO – International Organization for Standardization (Международная организация по стандартизации)

IoT – Internet of Things (интернет вещей)

LAI – Leaf Area Index (индекс листовой поверхности)

LSTM – Long Short-Term Memory (нейросеть с долгой краткосрочной памятью)

LULC – Land Use / Land Cover (землепользование и земельный покров)

ML – Machine Learning (машинное обучение)

MAE – Mean Absolute Error (средняя абсолютная ошибка)

MLP – Multilayer Perceptron (многослойный перцептрон)

MNDWI – Modified Normalized Difference Water Index (модифицированный нормализованный разностный водный индекс)

MODIS – Moderate Resolution Imaging Spectroradiometer (спектрорадиометр среднего разрешения NASA)

MSE – Mean Squared Error (среднеквадратичная ошибка)

NASA - National Aeronautics and Space Administration (Национальное управление по аэронавтике и исследованию космического пространства)

NDBI – Normalized Difference Built-up Index (нормализованный разностный индекс застройки)

NDMI – Normalized Difference Moisture Index (нормализованный разностный индекс влажности)

NDTI – Normalized Difference Tillage Index (нормализованный разностный индекс обработки почвы)

NDVI – Normalized Difference Vegetation Index (нормализованный разностный вегетационный индекс)

NDWI – Normalized Difference Water Index (нормализованный разностный водный индекс)

NN – Neural Network (нейронная сеть)

NTL – Night-Time Lights (данные ночного освещения)

OGC – Open Geospatial Consortium (Открытый геопространственный консорциум)

PDP – Partial Dependence Plot (график частичной зависимости)

R^2 – коэффициент детерминации

RBF – Radial Basis Function (радиальная базисная функция)

RF – Random Forest (алгоритм случайного леса)

RMSE – Root Mean Square Error (корень из среднеквадратичной ошибки)

RNN – Recurrent Neural Network (рекуррентная нейронная сеть)

SHAP – SHapley Additive exPlanations (аддитивные объяснения Шепли)

STD - Spatiotemporal data (пространственно-временные данные)

SVM – Support Vector Machine (метод опорных векторов)

SVR – Support Vector Regression (регрессия на опорных векторах)

UN GGIM (United Nations Global Geospatial Information Management - глобальное управление геопространственной информацией)

USGS - United States Geological Survey (Геологическая служба США)

UTM – Universal Transverse Mercator (универсальная поперечная проекция Меркатора)

VRS – Variable Returns to Scale (переменная отдача от масштаба)

WFS – Web Feature Service (веб-сервис доступа к векторным пространственным данным)

WMS – Web Map Service (веб-сервис картографических изображений)

XGBoost – Extreme Gradient Boosting (градиентный бустинг деревьев решений)

НОРМАТИВНЫЕ ДОКУМЕНТЫ

Закон Республики Казахстан «О геодезии, картографии и пространственных данных», от 21 декабря 2022 года № 166-VII ЗРК.

Об утверждении Концепции развития агропромышленного комплекса Республики Казахстан на 2021–2030 годы. Постановление Правительства Республики Казахстан от 30 декабря 2021 года № 960.

Об утверждении Концепции цифровой трансформации, развития отрасли информационно-коммуникационных технологий и кибербезопасности на 2023–2029 годы. Постановление Правительства Республики Казахстан от 28 марта 2023 года № 269.

Земельный кодекс Республики Казахстан. 20 июня 2003 года № 442-I ЗРК,

Лесной кодекс Республики Казахстан, от 8 июля 2003 года № 477 ЗРК.

Водный кодекс Республики Казахстан от 9 июля 2003 года № 481 ЗРК, основополагающий документ.

Экологический кодекс Республики Казахстан от 2 января 2021 года № 400-VI ЗРК.

Закон Республики Казахстан «О космической деятельности», с изменениями и дополнениями по состоянию на 11.04.2019 г. ЗРК.

Закон Республики Казахстан от 24 ноября 2015 года № 418-V ЗРК. «Об информатизации»

Национальный план развития Республики Казахстана до 2029 года. Астана, 2024 год. Указ Президента Республики Казахстан от 30 июля 2024 года № 611.

О науке и технологической политике. Закон Республики Казахстан от 1 июля 2024 года № 103-VIII ЗРК.

Об искусственном интеллекте. Закон Республики Казахстан от 17 ноября 2025 года № 230-VIII ЗРК.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	10
ГЛАВА 1. ТЕОРЕТИКО-МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ИНТЕГРАЛЬНОЙ ОЦЕНКИ АГРАРНОЙ ДЕЯТЕЛЬНОСТИ	13
1.1 Современное понимание устойчивого развития агропромышленного комплекса и сельских территории...	14
1.2 Аграрная инфраструктура пространственных данных 2.0 как основа цифровой трансформации агропромышленного комплекса и сельских территорий	17
1.3 Необходимость интегральной оценки государственной политики: экономические, социальные и экологические аспекты	20
ГЛАВА 2. ИСТОЧНИКИ ПРОСТРАНСТВЕННО- ВРЕМЕННЫХ И БОЛЬШИХ ДАННЫХ В АгроИПД 2.0....	22
2.1 Пространственные и временные характеристики данных	24
2.2 Основные источники пространственно-временных данных и Big Data для анализа аграрной деятельности.....	26
2.3 Роль ГИС и веб-технологий в обеспечении анализа данных	29
ГЛАВА 3. ПОДХОДЫ К ИНТЕГРАЛЬНОЙ ОЦЕНКЕ АГРАРНОЙ ДЕЯТЕЛЬНОСТИ	32
3.1. Характеристика исходного набора данных	32
3.2. Предварительная обработка данных: корреляционный анализ	35
3.3 Оценка агропроизводственной эффективности сельских округов методом Data Envelopment Analysis	42
ГЛАВА 4. МАШИННОЕ ОБУЧЕНИЕ ДЛЯ ОБЪЯСНЕНИЯ И ПРОГНОЗИРОВАНИЯ ИНДЕКСА DATA ENVELOPMENT ANALYSIS.....	50
4.1. Random Forest	50
4.2. Градиентный бустинг деревьев (последовательная аппроксимация сложных нелинейностей)	52
4.3. Регрессия на опорных векторах (SVR с RBF-ядром как гладкая регуляризованная аппроксимация)	54
4.4. Полносвязные нейронные сети (гибкость аппроксимации при повышенных требованиях к валидации).....	56

4.5. Elastic Net (линейный базис для отбора и коммуникации) и сводная аналитика результатов	56
ГЛАВА 5. ГЛУБОКОЕ ОБУЧЕНИЕ ДЛЯ ВЫЯВЛЕНИЯ ДРАЙВЕР-ФАКТОРОВ ИНДЕКСА DATA ENVELOPMENT ANALYSIS	62
5.1. «Компактные» и «глубокие» полносвязные сети в TensorFlow (SimpleTF / DeepTF).....	62
5.2. Резидуальные сети (ResidualNet) как средство стабилизации глубоких представлений.....	64
5.3. Внимание и самовнимание (AttentionNet) для захвата контекстных зависимостей.....	65
5.4. Рекуррентные сети (LSTMNet) для использования панельной временной структуры.....	67
5.5. Классические MLP-архитектуры (SimpleNet/DeepNet) и сравнительный синтез результатов	68
ГЛАВА 6. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПОДХОДОВ И ИНТЕГРАЦИЯ РЕЗУЛЬТАТОВ.....	72
6.1 Сравнительные характеристики эмпирического, ML и DL подходов	73
6.2 Комбинированные модели и возможности гибридных решений	77
6.3 Практические аспекты интеграции и использование результатов.....	81
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	86
ГЛОССАРИЙ.....	94
ПРИЛОЖЕНИЕ 1 Название сельских округов.....	97
ПРИЛОЖЕНИЕ 2 Индексы DEA по сельским округам.....	100
ПРИЛОЖЕНИЕ 3. Программная реализация методики (скрипт на языке Python).....	108

ВВЕДЕНИЕ

Настоящий труд направлен на разработку методических подходов для интегральной оценки развития агропромышленного комплекса (АПК) и сельских территорий (СТ) на примере пространственно-временных, социально-экономических и экологических данных 84 сельских округов Оskarовского, Бухар-Жырауского, Абайского и Каркаралинского районов Карагандинской области (название сельских округов приведены в приложении 1), исходные сведения о которых приведены в [1].

Устойчивое развитие АПК и СТ представляет собой многомерную задачу, требующую координации экономических, социальных и экологических аспектов в условиях растущего спроса на продовольствие и ограниченности природных ресурсов [2,3]. По прогнозам ООН, мировое население к 2050 году достигнет 9,7 млрд человек, что усилит нагрузку на сельское хозяйство и потребует повышения эффективности производства без ущерба для окружающей среды и качества жизни сельских сообществ [4]. Параллельно с этим изменение климата, деградация почв и истощение водных ресурсов создают дополнительные риски, что делает необходимым внедрение оперативных методов мониторинга и оценки аграрных практик на разных территориальных уровнях [5].

Ведущие аграрные державы - Соединённые Штаты, Канада и государства Европейского союза - уже интегрировали технологии дистанционного зондирования, беспилотные летательные аппараты и облачные ГИС-платформы для наблюдения за динамикой культур и состояния земель в режиме реального времени [6–8]. В Китае и Индии действуют мобильные решения, ориентированные на мелких и средних фермеров, что повышает их вовлечённость в рациональное землепользование и способствует снижению социального неравенства на селе [9,10]. Аналогичные инициативы реализуются в Бразилии и России, где национальные программы поддержки аграриев включают развитие интегрированных систем учёта и аналитики пространственных данных [11,12].

Несмотря на технические достижения, отсутствует единый методический каркас для комплексной оценки последствий аграрной деятельности. Традиционные показатели эффективности - валовые сборы, экспортные доходы, уровень занятости - не дают полной

картины воздействия на экосистему и социальные структуры [13]. Чтобы преодолеть этот разрыв, необходима синергия нескольких подходов: стандартизированная «Аграрная инфраструктура пространственных данных 2.0» (АгроИПД 2.0) должна обеспечивать совместимость источников через протоколы WMS/WFS (WMS - Web Map Service, веб-сервис картографических изображений; WFS - Web Feature Service, веб-сервис доступа к векторным пространственным данным) и метаданные ISO 19115 (ISO - International Organization for Standardization, Международная организация по стандартизации) [14,15], а аналитическая платформа - объединять методы непараметрического DEA (Data Envelopment Analysis - анализ оболочки данных) и современные алгоритмы машинного обучения (Random Forest, BiLSTM (Bidirectional Long Short-Term Memory - двунаправленная рекуррентная нейросеть с долгой краткосрочной памятью), XGBoost (Extreme Gradient Boosting - экстремальный градиентный бустинг деревьев решений)) для многокритериальной оценки и прогнозирования [16–18].

Интегральная оценка предполагает работу с «5V» больших данных (volume - объём, variety - разнообразие, velocity - скорость, veracity - достоверность, value - ценность), что требует не только вычислительных мощностей, но и выверенных процедур предобработки, очистки и валидации данных [19]. Выявление скрытых закономерностей и взаимосвязей между политическими мерами и результатами в агросекторе обеспечивается внедрением гибридных моделей, способных учитывать «нежелательные» явления (эрозию почв, выбросы парниковых газов) и социальные эффекты (миграцию, занятость) одновременно [20].

Цель настоящей рекомендации - предложить унифицированные методические подходы для разработки и эксплуатации АгроИПД 2.0 как функции государственной и корпоративной политики в аграрной отрасли.

Практическая значимость состоит в том, что органы управления получат инструментарий для:

- подготовки оперативных отчётов с пространственно-временным анализом ключевых показателей;
- обоснования корректировок программ субсидирования и инвестиций на основе прогнозных сценариев;

- оценки долгосрочных последствий эколого-социальных рисков и разработке мер по их снижению;

- обмена лучшими практиками и сравнительного анализа эффективности аграрной деятельности между регионами и странами.

Реализация рекомендаций позволит повысить прозрачность принятия решений, сократить неоправданные затраты и ускорить достижение национальных и глобальных Целей устойчивого развития к 2030 году.

При необходимости, заинтересованный читатель может получить все скрипты в Word формате, написав письмо на следующий E-mail: Alipbeki.Onggarbek@kaznu.kz

Глава 1. ТЕОРЕТИКО-МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ИНТЕГРАЛЬНОЙ ОЦЕНКИ АГРАРНОЙ ДЕЯТЕЛЬНОСТИ

Современные аграрные системы испытывают давление со стороны климатических изменений, связанных с ним деградации почв и растущей потребности в продовольствии. В этих условиях концепция устойчивого развития АПК и СТ перестаёт быть абстрактной идеей и превращается в практический ориентир для государственной политики: она объединяет повышение производительности, социальную защиту сельского населения и сохранение экосистем [2,3].

Геоинформационные технологии и системы пространственных данных становятся неотъемлемой основой управления аграрной отраслью. «Аграрная инфраструктура пространственных данных 2.0» включает вебориентированные сервисы WMS/WFS и стандарты ISO 19115, что обеспечивает единство форматов и прозрачность обмена информацией между республиканскими, региональными и коммерческими структурами [14]. Спутниковое дистанционное зондирование, мониторинг с беспилотных летательных аппаратов (БПЛА) и наземные датчики сенсоры и интернет вещи (Internet of Things - IoT) создают мультиуровневую картину состояния посевных площадей, водных ресурсов и биоразнообразия, позволяя анализировать динамику изменений в пространственно-временном разрезе без потери деталей.

Точность и своевременность данных - лишь предпосылка. Для преобразования информации в управленческие решения необходимы интегральные методики оценки. Комбинация непараметрического DEA, позволяющего оценить относительную эффективность хозяйств, и машинного и глубокого обучения, способного выявлять сложные нелинейные зависимости, создаёт прочный аналитический каркас [16,17]. Экономические показатели (урожайность, рентабельность), социальные метрики (уровень занятости, доступность социальной инфраструктуры) и экологические параметры (фоновая нагрузка на почвы и водоёмы) объединяются в общую модель, чтобы обеспечить многокритериальный анализ и прогнозирование результатов аграрных программ.

Интегральная оценка аграрной деятельности призвана обеспечить обоснованность субсидий, грантов и инфраструктурных вложений, минимизировать риски экологических и социальных сбоев и повысить адаптивность управления к локальным и глобальным вызовам. Именно синтез передовых методов анализа и устойчивых принципов развития создаёт основу для перехода к «умному» агробизнесу, где решения принимаются на основе объективных данных и системных сценариев.

1.1 Современное понимание устойчивого развития агропромышленного комплекса и сельских территорий

Устойчивое развитие АПК и СТ в XXI веке – это не просто сбалансированное сочетание экономических, экологических и социальных факторов. Это управляемая, количественно измеримая и пространственно-определённая система, способная адаптироваться к многовекторным вызовам: климатическим, технологическим, демографическим и политическим. Современные международные подходы рассматривают устойчивость не как статическое состояние, а как динамический процесс, который требует постоянного мониторинга, анализа и предиктивного управления в реальном времени [2].

Ключевое отличие современного понимания устойчивости заключается в признании территориальной и временной неоднородности аграрных процессов. Один и тот же политический инструмент - например, субсидии на удобрения - может давать противоположные эффекты в разных регионах, в зависимости от ресурсной базы, климатических условий, плотности инфраструктуры и уровня цифровой зрелости специалистов сельского хозяйства. Именно поэтому в новых системах управления устойчивым развитием приоритет отводится пространственно-временным данным (Spatiotemporal data - STD), объединяющим показатели агрофизики, экономики, демографии и экологии с географической и временной привязкой [3].

Организация Объединённых Наций через Цели устойчивого развития (ЦУР), а также Всемирный банк, FAO (Food and Agriculture Organization - Продовольственная и сельскохозяйственная организация ООН) и Европейская комиссия подчёркивают, что основа

аграрной устойчивости – это способность адаптироваться к изменениям, не теряя при этом продуктивности и социальной стабильности. Такая способность в настоящее время определяется доступом к данным, технологическим возможностям их обработки и применением интеллектуальных аналитических платформ [5]. Пространственные данные позволяют визуализировать и анализировать эволюцию таких параметров, как деградация почв, распределение водных ресурсов, плотность занятости или миграционные потоки в сельской местности, и делают возможным ранжирование регионов по степени устойчивости и уязвимости [11].

Примером может служить опыт ЕС по внедрению так называемых Farm Sustainability Tools (FaST), которые собирают данные от сельхозпроизводителей, метеостанций и спутниковых платформ и автоматически рассчитывают показатели, отражающие устойчивость производства по ключевым направлениям. Используемые метрики включают такие параметры, как количество дней водного стресса, индекс разнообразия культур, динамика органического вещества в почве и уровень цифровизации хозяйства [21].

В рамках цифровой трансформации аграрной отрасли такие инструменты получают дополнительную функциональность через алгоритмы машинного и глубокого обучения, основанного на искусственном интеллекте. Современные исследования демонстрируют высокую эффективность гибридных моделей на основе Random Forest и Bi-LSTM при анализе многолетних пространственно-временных рядов, например, для прогнозирования урожайности, выявления латентных экологических угроз и оценки рисков продовольственной нестабильности [17].

Аналогичный подход реализуется в Казахстане. Формируемая в республике «Аграрная инфраструктура пространственных данных 2.0» (АгроИПД 2.0) обеспечивает многослойную интеграцию картографических, агротехнологических и социально-экономических данных в единую аналитическую систему. Это позволяет перейти от ведомственного планирования к комплексному межатраслевому управлению СТ, где учитываются как биофизические параметры, так и факторы, связанные с качеством жизни, доступом к услугам, миграцией и образованием [22].

Экологическая составляющая устойчивости также больше не оценивается обособленно. В новых подходах она интегрирована в общие модели оценки эффективности деятельности: например, выбросы парниковых газов или снижение биоразнообразия рассматриваются как «нежелательные выходы» (Undesirable outputs) и включаются в оценки эффективности хозяйств по методикам Data Envelopment Analysis (DEA) и её модификациям (Super-DEA, Malmquist Index) [11,20,23]. Это позволяет учитывать не только количественные, но и качественные эффекты реализации аграрных программ.

Неотъемлемым компонентом современного понимания устойчивости АПК и СТ становится устойчивость управления: наличие цифровой инфраструктуры, компетентных пользователей данных на местах, возможность постоянной валидации информации и прозрачности принятия решений. Устойчивость больше не воспринимается как сугубо экологическая категория - она становится характеристикой всей системы агроуправления.

Таким образом, современное понимание устойчивого развития АПК и СТ основано на следующих принципах:

- пространственно-временная детализированность всех процессов и решений;
- цифровая наблюдаемость и интероперабельность данных;
- интеграция экономических, социальных и экологических измерений в единые модели;
- использование интеллектуальных аналитических систем для предиктивного управления;
- институциональная способность к внедрению и адаптации решений на местах.

Такой подход создаёт фундамент для выстраивания современной аграрной деятельности, где устойчивость перестаёт быть абстрактной целью и становится результатом управляемого на основе цифрового обеспечения и территориально обоснованного процесса.

1.2 Аграрная инфраструктура пространственных данных 2.0 как основа цифровой трансформации агропромышленного комплекса и сельских территорий

Современные аграрные системы характеризуются высокой динамичностью, чувствительностью к климатическим и экономическим факторам, а также территориальной неоднородностью. В таких условиях цифровая трансформация АПК невозможна без полноценной пространственно-временной поддержки. Именно поэтому в последние годы всё большее значение приобретает аграрная инфраструктура пространственных данных нового поколения - АгроИПД 2.0, представляющая собой не просто хранилище данных, а функциональную основу для оперативного, аналитического и стратегического управления сельским хозяйством.

Трансформация систем принятия решений в агросекторе требует перехода от фрагментарных и ведомственно разрозненных баз к единой, открытой, стандартизированной и масштабируемой инфраструктуре. АгроИПД 2.0 проектируется как интеграционная платформа, объединяющая картографическую, агроэкологическую, климатическую, кадастровую, демографическую и экономическую информацию, привязанную к конкретным координатам и временным промежуткам. На сегодняшний день такие системы активно внедряются в странах ЕС в рамках реализации директивы INSPIRE (Infrastructure for Spatial Information in Europe - инфраструктура пространственной информации Европейского союза), а также в Канаде, США, Бразилии и Индии как основа агроэкосистемной политики, а также точного и умного сельского хозяйства [11,23].

В международной практике уже устоялось понимание, что пространственно-временные данные — это не просто вспомогательный инструмент, а ядро любой системы устойчивого агроуправления. Например, в докладах Европейской комиссии подчёркивается, что именно доступ к корректным, актуальным и геопривязанным данным позволяет фермерам адаптироваться к климатическим рискам, а государственным органам - объективно планировать агроподдержку и реагировать на локальные угрозы, такие как засухи, эрозия почв или отток трудовых ресурсов [3].

В Республике Казахстан формирование АгроИПД 2.0 является частью национальной стратегии цифровизации аграрной отрасли. В официальных рекомендациях Министерства сельского хозяйства подчёркивается, что система должна обеспечивать не только хранение, но и автоматическую обработку и публикацию информации в форме геосервисов, совместимых с международными стандартами OGC (Open Geospatial Consortium - Открытый геопространственный консорциум) и ISO (International Organization for Standardization). Такой подход позволит объединить в единое информационное пространство все основные массивы пространственных данных: от карт границ земельных участков и цифровых моделей рельефа до данных спутникового мониторинга урожайности и социальных показателей сельских территорий [24].

Особое внимание в архитектуре АгроИПД 2.0 уделяется *интероперабельности*, то есть способности системы интегрироваться с другими цифровыми платформами. Это позволяет реализовать сквозной обмен данными между кадастровыми, налоговыми, статистическими, климатическими и агрометеорологическими системами. Таким образом, государственные органы, сельхозпроизводители и исследовательские центры могут использовать одни и те же данные, адаптированные под свои задачи, без дублирования и потерь в актуальности.

По данным Международного института цифровой трансформации сельского хозяйства, более 60 % аграрных решений в странах с развитой цифровой инфраструктурой сегодня принимаются на основе пространственного анализа, включая такие направления как прогноз урожайности, оценка плодородия, планирование орошения, борьба с вредителями и анализ рисков. Причём эффективность таких решений возрастает в разы, когда данные интегрируются в единую геоинформационную систему, а не используются в изолированном виде [25].

АгроИПД 2.0 также закладывает основу для создания цифровых двойников сельских территорий и агропредприятий. Это означает моделирование всего агро-пространственного комплекса в цифровой среде, с учётом не только физических параметров (почва, климат, ландшафт), но и инфраструктурных и социальных характеристик. Такая модель позволяет проводить сценарный анализ: например, как изменится продуктивность территории при сокращении осадков

на 10 %, или как влияет замедление миграции на занятость в селе. По сути, это делает возможным переход от *реактивного* к *проактивному* типу управления, где политика формируется не по факту, а на основе прогнозов и смоделированных эффектов.

Мировой опыт показывает, что ключевым элементом успеха ИПД становится их открытость и доступность. Например, в Нидерландах система AgroDataCube предоставляет разработчикам и агробизнесу API-доступ (API - Application Programming Interface, программный интерфейс приложения) ко всем основным слоям данных - от истории посевов до уровня грунтовых вод. В Канаде реализация инициативы Open Government Data дала возможность каждому фермеру свободно использовать геоинформационные ресурсы без сложных согласований. Казахстан движется в том же направлении: предполагается внедрение системы гранулярных уровней доступа, позволяющей сочетать открытые и конфиденциальные сегменты информации с учётом интересов государства и бизнеса [26].

В то же время формирование АгроИПД невозможно без параллельного развития кадровой и нормативной базы. Необходимы специалисты, способные управлять ГИС-инфраструктурой (ГИС - геоинформационные системы), работать с аналитическими инструментами и обеспечивать техническую поддержку пользователей. Также требуется правовое закрепление понятий, регламентов доступа, механизмов финансирования и ответственности за качество данных. Международные рекомендации, в частности отчёт United Nations GGIM (Global Geospatial Information Management - глобальное управление геопространственной информацией), подчёркивают необходимость синхронного развития технической, институциональной и образовательной составляющих пространственных систем [27].

Так, АгроИПД 2.0 становится не просто цифровым хранилищем, а стратегическим элементом цифровой трансформации аграрной отрасли. Она позволяет перейти от бумажного или ведомственно-ориентированного управления к пространственно-интегрированному управлению, основанному на объективных данных и предиктивной аналитике. Это особенно важно в условиях усиливающейся климатической нестабильности, давления на ресурсы и необходимости адаптивной аграрной политики.

1.3 Необходимость интегральной оценки государственной политики: экономические, социальные и экологические аспекты

Оценка государственной политики в агропромышленной сфере традиционно опиралась на набор экономических показателей: валовой выпуск продукции, уровень инвестиций, динамику экспорта и импортозамещения. Однако в современном контексте этого уже недостаточно. Эффективное развитие сельских территорий и аграрной экономики требует системного подхода, в котором наряду с экономическими результатами учитываются также экологические последствия и социальные эффекты. Именно поэтому всё больше стран переходят к интегральной системе оценки аграрной политики, охватывающей многомерные аспекты и основанной на пространственно-временных данных.

Такой подход становится особенно актуальным в условиях усиливающейся климатической изменчивости, водного дефицита, деградации земель и снижения численности сельского населения. Он позволяет не только оценивать текущие эффекты принимаемых решений, но и прогнозировать отложенные последствия. Ключевым условием этого является наличие инфраструктуры пространственных данных, развитие которой отмечено как стратегический приоритет в рекомендациях таких организаций, как FAO и World Bank [3].

Интегральная оценка позволяет объединить в одной аналитической системе как показатели агроэкономики (производительность, прибыльность, экспортный потенциал), так и параметры, связанные с социальной устойчивостью (занятость, уровень жизни, миграция, доступ к инфраструктуре), а также экологическую устойчивость (изменения в использовании земли, выбросы, плодородие почв, состояние водных ресурсов). Примеры такого системного анализа представлены в работах Wu et al. (2022) и Saidmamatov et al. (2023), где продемонстрировано, как комплексные показатели могут применяться для оценки государственной политики с межрегиональными и межвременными различиями [28,29].

При этом не требуется погружение в отдельные математические модели. Существующий арсенал аналитических подходов

(эконометрические, пространственные, многокритериальные) уже проверен на практике в различных странах, и может быть адаптирован к условиям конкретного региона. Важно то, что интеграция этих методов позволяет уходить от фрагментарной оценки и переходить к целостному представлению об эффективности аграрной политики - с учётом локальных условий, вовлечения акторов и возможных компромиссов между развитием и устойчивостью.

Пространственно-временной характер аграрных процессов требует также соответствующего аналитического инструментария. Наличие высокочастотных наблюдений (например, по спутниковым снимкам, данным о землепользовании, погоде, урожайности) создаёт основу для регулярного мониторинга изменений. Это особенно ценно для оценки тех аспектов, которые невозможно выявить статистически за короткий период - например, деградации земель, утраты водных ресурсов или сокращения численности сельского населения.

В этой связи всё более важным становится вопрос институциональной готовности к использованию интегральных оценок. Речь идёт не только о технических инструментах, но и о политической воле, о готовности применять данные в принятии решений, о механизмах межведомственного взаимодействия. Как отмечается в ряде международных исследований, высокая доступность данных и развитие платформ открытого доступа являются необходимым условием формирования прозрачной и подотчётной государственной политики [30,31].

Таким образом, интегральная оценка аграрной политики сегодня — это не просто инструмент анализа. Это институциональный механизм, позволяющий связать между собой ЦУР, пространственное планирование, ресурсное управление и социальную политику. Именно в этом контексте важна интеграция современных аналитических подходов - без необходимости делать ставку на одну конкретную модель, но с пониманием, что решения должны базироваться на проверяемых, сопоставимых и пространственно-ориентированных данных.

Глава 2. ИСТОЧНИКИ ПРОСТРАНСТВЕННО-ВРЕМЕННЫХ И БОЛЬШИХ ДАННЫХ ДЛЯ АГРОИПД 2.0

Развитие АгроИПД 2.0 невозможно без глубокого понимания источников и характеристик информации, на которой она основана. В контексте цифровизации сельского хозяйства ключевую роль играют пространственные, временные и статистические данные (в целом Big Data), которые формируют основу для мониторинга, прогнозирования и анализа состояния аграрной и сельской среды. Эти данные не только обеспечивают основу для картографических приложений и аналитических платформ, но и определяют качество принимаемых решений в области государственной аграрной политики и управления сельскими территориями.

Пространственные данные поступают из широкого спектра источников. Наиболее значимыми из них являются данные дистанционного зондирования Земли (ДЗЗ) - спутниковые снимки различного разрешения (например, Planet, Sentinel, Landsat, MODIS (Moderate Resolution Imaging Spectroradiometer – спектро-радиометр среднего разрешения)), а также данные, полученные с беспилотных летательных аппаратов и наземных геодезических комплексов. Данные такого рода позволяют оценивать структуру землепользования, мониторить вегетацию, почвенно-гидрологическое состояние, динамику водоёмов и другие пространственно распределённые явления [3,32].

Согласно рекомендациям Продовольственной и сельскохозяйственной организации ООН, внедрение цифровых платформ с привязкой к пространственным данным рассматривается как ключевой шаг к устойчивому управлению продовольственными системами (FAO, 2022) [3]. Принципы открытых пространственных данных, поддерживаемые такими международными инициативами, как Group on Earth Observations (GEO) и Open Geospatial Consortium (OGC), способствуют стандартизации форматов и упрощению трансграничного обмена агроэкологической информацией [33,34].

Наряду с пространственными, важнейшую роль играют временные характеристики данных, определяющие повторяемость и актуальность измерений. Регулярные спутниковые наблюдения (например, каждые 5 дней по Sentinel-2) позволяют создавать

динамические модели изменения состояния посевов, влажности почв и водного баланса, а также выявлять отклонения от нормы. Хронология данных становится особенно ценной при построении моделей урожайности, агроклиматического риска и выявления последствий политических интервенций.

К пространственным источникам подключаются большие данные (Big Data), формируемые из административной, экономической и социальной статистики. В условиях Республики Казахстан важнейшим поставщиком такой информации выступает Бюро национальной статистики, публикующее данные в разрезе областей, районов, и сельских округов. Они включают аграрную производственную отчетность, занятость, миграцию, структуру хозяйств, показатели по инвестированию и государственным субсидиям. Именно эти данные связывают пространственный анализ с социально-экономическим измерением, формируя интегральную платформу для управления развитием сельских территорий [35,36].

В совокупности пространственные и статистические данные образуют многоуровневую систему наблюдений, которая при надлежащей организации способна стать не просто инструментом мониторинга, а полноценной базой для цифровых двойников сельской территории, применимых в агрополитике, землепользовании и климатическом планировании.

Один из ключевых вызовов - интеграция разнородных источников данных. Данные ДЗЗ поставляются в формате георастров, данные GPS (Global Positioning System - глобальная система позиционирования) - в векторном виде, статданные – в цифровой табличной форме, а административная статистика требует пространственной привязки к кадастровым единицам. Международные стандарты (например, ISO 19115 для описания метаданных, INSPIRE в ЕС, OGC WMS/WFS для картографических сервисов) дают необходимые технические решения для обеспечения совместимости, но требуют соответствующей цифровой инфраструктуры и кадровой поддержки [15].

Таким образом, источники пространственно-временных и больших данных в АгроИПД 2.0 представляют собой многоуровневую, интегративную экосистему, в которой каждый компонент имеет свою специфику, но только в совокупности они дают

аналитическую полноту. В последующих разделах главы 2 рассматриваются ключевые характеристики этих данных, основные платформы, используемые в аграрной практике, и роль геоинформационных и веб-технологий в их обработке и интерпретации.

2.1 Пространственные и временные характеристики данных

Пространственные и временные данные играют ключевую роль в цифровой трансформации АПК. Их качество, совместимость и согласованность напрямую определяют эффективность аналитических систем, в том числе в рамках АгроИПД. Именно поэтому критически важно обеспечить единые принципы их сбора, хранения, обработки и визуализации.

Пространственные характеристики включают в себя географическую привязку объектов, координатные системы (например QazTRF-2023, WGS-84 (World Geodetic System 1984 - всемирная геодезическая система координат 1984 года), типы данных (векторные или растровые) и масштабность представления. Геоданные могут представлять собой точечные объекты (например, метеостанции), линейные элементы (ирригационные каналы, дороги) или полигоны (поля, пастбища, административные границы) [37].

Временные характеристики данных связаны с периодичностью наблюдений, временным разрешением и актуальностью информации. Например, спутниковые данные программы Sentinel обновляются каждые 5–10 дней, в то время как статистические отчеты могут публиковаться раз в квартал или год. Для динамичного управления сельским хозяйством необходимо наличие как ретроспективных, так и оперативных временных данных, с возможностью их корректного совмещения.

Одной из главных проблем в аграрной аналитике является неконсистентность пространственно-временных данных, поступающих из различных источников. Это касается различий в форматах файлов, системах координат, единицах измерения, а также отсутствии или несовпадении метаданных. Например, спутниковые снимки как данные GPS о местоположении техники могут быть представлены в WGS84, а муниципальная статистика - в административных кодах, не привязанных к геометрии.

Для решения этой проблемы в АгроИПД применяется единая система стандартизации, включая:

- использование международных стандартов метаданных ISO 19115 [15];
- обязательную географическую привязку всех данных;
- унификацию единиц измерения (гектары, тонны, мм осадков и т.д.);
- синхронизацию временных интервалов (например, пересчет данных до единой временной шкалы: неделя, месяц, сезон) [38].

Дополнительно применяется технология ETL (extract-transform-load), позволяющая извлекать данные из различных источников (спутники, дроны, сенсоры, статистика), трансформировать их в согласованный формат и загружать в единое хранилище. Это обеспечивает пространственно-временное соответствие, без которого невозможен полноценный кросс-анализ [27].

Интеграция данных невозможна без наличия каталогов и реестров метаданных, в которых указывается происхождение, точность, дата обновления, формат, проекция и доступность каждого набора данных. Это позволяет пользователям - от фермеров до министерств - быстро ориентироваться в массиве информации и применять её для принятия решений [33].

Важно отметить, что пространственно-временные данные охватывают разные масштабы анализа:

- микроуровень (поле, хозяйство) - данные с сенсоров, дронов, агрономических обследований;
- мезоуровень (район, область) - агростатистика, спутниковые индексы;
- макроуровень (страна, регион) - данные международных программ мониторинга (например, FAOSTAT (статистическая база данных FAO), Copernicus).

В условиях ускоряющихся климатических изменений, нестабильности рынков и трансформации аграрной политики наличие согласованных пространственно-временных данных становится не просто техническим требованием, а стратегическим ресурсом. Их корректная обработка позволяет строить карты продуктивности, прогнозировать урожайность, моделировать сценарии развития территорий и оценивать эффективность политик устойчивого развития.

2.2 Основные источники пространственно-временных данных и Big Data для анализа аграрной деятельности

В условиях стремительного роста объема данных, связанных с функционированием АПК, цифровая трансформация в сельском хозяйстве опирается на два ключевых типа информации – пространственно-временные данные (ПВД) и большие данные (Big Data). Их интеграция в рамках АгроИПД 2.0 позволяет формировать целостную аналитическую инфраструктуру, обеспечивающую количественную оценку и прогноз эффективности государственной аграрной политики.

Пространственные данные представляют собой информацию, содержащую координатную привязку объектов к конкретной местности. К основным источникам ПВД в агрополитике относятся:

- данные дистанционного зондирования Земли. Крупные международные спутниковые программы, такие как *Copernicus* (Sentinel-1, -2) и *Landsat* (NASA-National Aeronautics and Space Administration - Национальное управление по аэронавтике и исследованию космического пространства/USGS (United States Geological Survey - Геологическая служба США), обеспечивают регулярное получение изображений с пространственным разрешением от 10 до 30 метров и временной частотой от 5 до 16 дней. Они позволяют определять структуру посевов, границы полей, вегетационные индексы (NDVI - Normalized Difference Vegetation Index, нормализованный разностный вегетационный индекс; EVI - Enhanced Vegetation Index, улучшенный вегетационный индекс), а также выявлять деградацию почв, засухи и риски продовольственной безопасности [39].

- данные от беспилотных летательных аппаратов (БПЛА) и наземных сенсоров, снабженных IoT (Internet of Things). Их используют на микроуровне (объект, поле, хозяйство) для точного картирования урожайности, агрохимического состава почв и метеоусловий. Эти данные обладают высокой детализацией, но ограничены по охвату и требуют локальной калибровки.

- геоинформационные и кадастровые системы, где содержатся сведения о границах сельхозугодий, праве собственности, категориях землепользования, зонах мелиорации и т. д. Их значение

особенно велико для мониторинга земельной политики и анализа целевого использования государственных субсидий.

- IoT-устройства (Internet of Things), такие как датчики температуры, влажности, уровня pH почвы, датчики в зернохранилищах и на технике. Они формируют непрерывные временные ряды с высокой частотой наблюдения и служат основой для систем предиктивной аналитики, точного и умного сельского хозяйства.

Параллельно с ПВД, агрополитика опирается на массивы больших данных, полученных из статистических, административных и транзакционных источников. В Республике Казахстан такими источниками являются:

- Бюро национальной статистики - предоставляет данные по производству сельхозпродукции, численности сельского населения, занятости, доходам и затратам домохозяйств, доступу к инфраструктуре, сельской миграции и т. д. Эти данные обладают высокой формализованностью и территориальной привязкой (области, районы, сельские округа);

- Министерство сельского хозяйства, Министерство искусственного интеллекта и цифрового развития, Министерство национальной экономики и Министерство финансов - источники данных по государственным программам (например, «Ауыл – ел бесігі»), распределению субсидий, грантов, лизингов, а также по инвестициям в АПК, сельскую инфраструктуру и инновации;

- Казгидромет и международные метеослужбы - поставщики временных рядов по температуре, осадкам, индексам засушливости, агроклиматическим зонам и др. Эти данные критичны для оценки воздействия климатических факторов на эффективность аграрной политики;

- Агроплатформы и CRM (Customer Relationship Management) - системы в АПК - коммерческие и государственные цифровые сервисы, где регистрируются закупки, продажи, логистика, структура затрат фермеров. Это обеспечивает транзакционную аналитику в разрезе субъектов хозяйствования.

На международном уровне доступны такие источники, как:

- FAOSTAT - глобальная статистика по продовольствию, рынкам, политике и ресурсам [39];

- GEOGLAM (GEO Global Agricultural Monitoring - глобальный сельскохозяйственный мониторинг GEO) - мониторинг урожайности в рамках глобальной инициативы ООН [40];

- World Bank Open Data & Agricultural Observatory – данные по инвестициям, устойчивости, цифровизации сельского хозяйства [41].

Несмотря на большое количество источников, их интеграция осложнена рядом факторов:

- разнородность форматов - от GeoTIFF и HDF5 (для снимков) до XLSX, CSV и API-интерфейсов;

- неодинаковая пространственная и временная детализация - спутниковые данные имеют 10-метровое разрешение, тогда как статистика публикуется в разрезе районов;

- отсутствие единых идентификаторов - невозможно напрямую связать субъект хозяйствования с конкретным полем, не имея геокодированной базы;

- разная периодичность обновления - от ежечасных показателей метеостанций до годовых сводок о производительности.

Для решения этих задач в АгроИПД применяются принципы Extract–Transform–Load, пространственное геокодирование, перевод в открытые форматы и соблюдение стандартов метаданных [15].

Собранные данные позволяют:

- анализировать, как инвестиции в сельскую инфраструктуру влияют на урожайность и занятость;

- оценивать эффективность субсидий в зависимости от характеристик земель и хозяйств;

- моделировать климатические и экономические риски;

- обеспечивать справедливое распределение государственных ресурсов по регионам.

Таким образом, источники ПВД и Big Data формируют фундамент аграрной аналитики нового поколения - основанной на пространственной точности, временной синхронности и многомерной интерпретации процессов.

2.3 Роль ГИС и веб-технологий в обеспечении анализа данных

Современные вызовы в сельском хозяйстве - от изменения климата до нестабильности рынков и деградации земель - требуют использования цифровых инструментов, обеспечивающих высокоточную диагностику и принятие решений на основе данных. ГИС и веб-технологии выступают ключевыми компонентами этой цифровой трансформации, играя роль как хранилища, так и платформы для анализа пространственно-временных данных.

Геоинформационные системы представляют собой интеграционные платформы, предназначенные для сбора, хранения, анализа, интерпретации и визуализации пространственных данных. В сельском хозяйстве они позволяют отображать и анализировать:

- распределение посевных площадей;
- изменение почвенных характеристик;
- индексы растительности (NDVI, EVI);
- метеорологические и климатические условия;
- административные границы и др.

Использование ГИС даёт возможность не только визуализировать территориальные различия, но и интегрировать данные разного типа и источника - от спутниковых изображений до кадастровой и статистической информации. Это становится особенно актуальным при разработке аграрной политики, планировании субсидий, мониторинге программ устойчивого землепользования и поддержке фермерских решений [37].

С переходом к онлайн-сервисам на первый план выходит концепция веб-ГИС - геоинформационных систем, доступ к которым осуществляется через веб-браузер. Это позволяет:

- использовать данные в реальном времени;
- работать без установки специализированного ПО;
- взаимодействовать с данными на различных уровнях (национальном, региональном, локальном);
- создавать пользовательские интерфейсы и дашборды с картографическими слоями.

Применение веб-ГИС особенно эффективно в задачах пространственного анализа для органов государственной власти,

сельхозпроизводителей, агрономов и исследователей. Например, фермер может через смартфон проверить данные по влажности почвы, тогда как администрация сельского округа анализирует распределение государственных субсидий на интерактивной карте, сопоставляя это с фактическими урожаями.

Ключевым элементом ГИС-аналитики является соблюдение стандартов совместимости. В международной практике используются:

- OGC WMS/WFS/WMTS (WMTS - Web Map Tile Service, веб-сервис тайловых картографических слоёв) - протоколы передачи картографических слоёв;
- ISO 19115 - описание метаданных геоинформационных ресурсов;
- INSPIRE (EC) - единая система классификации и согласования пространственных данных для государственного управления;
- GeoJSON (JavaScript Object Notation - Объектная нотация JavaScript), GML (Geography Markup Language - язык географической разметки), KML (Keyhole Markup Language - язык разметки геоданных Keyhole) - форматы хранения геоданных с возможностью обмена между системами [33].

Соблюдение этих стандартов гарантирует, что данные, собранные в одном регионе или приложении, смогут быть интегрированы в единую систему пространственного анализа на государственном или транснациональном уровне.

Примеры практического применения ГИС и веб-ГИС включают:

- мониторинг урожайности по спутниковым данным (NDVI, LAI (Leaf Area Index - индекс листовой поверхности)) в сочетании с отчётами хозяйств;
- создание геопорталов для визуализации мер государственной поддержки на карте (например, World Bank Agricultural Geoportal);
- оценка продовольственной безопасности в рамках FAO GeoNetwork и WIEWS;
- управление пастбищами и профилактика деградации земель на основе зональных моделей.

По данным FAO, около 70 стран мира используют ГИС-инструменты в национальном аграрном планировании, и более 30 из них - в формате веб-сервисов с открытым доступом [42].

Для Казахстана веб-ГИС представляют особую ценность благодаря территориальной протяжённости, разнообразию природных зон и важности удалённого мониторинга в регионах с низкой плотностью населения. Развитие геопорталов (например, на базе GeoServer и PostGIS), поддержка открытых платформ и обучение специалистов в области пространственного анализа - важнейшие условия формирования устойчивой инфраструктуры данных в АПК.

Глава 3. ПОДХОДЫ К ИНТЕГРАЛЬНОЙ ОЦЕНКЕ АГРАРНОЙ ДЕЯТЕЛЬНОСТИ

3.1. Характеристика исходного набора данных

Исходной информационной базой исследования служит панель наблюдений «округ–год»: одни и те же сельские административно-территориальные единицы описываются ежегодно на протяжении 15 последовательных лет. Всего сформировано 1260 записей, каждая из которых содержит 48 переменных: 46 количественных показателей и два служебных поля, обеспечивающих идентификацию округа и привязку к году [1].

Такая организация данных даёт два принципиальных преимущества. Явная временная метка (Year) упорядочивает наблюдения и позволяет отделять изменения во времени от устойчивых различий между территориями. Привязка каждой записи к конкретному сельскому округу, в свою очередь, открывает возможность интеграции с ГИС-данными и картографического анализа. Именно поэтому пространственно-временная панель стала стандартом для анализа сельских территорий: она одновременно улавливает «эффект места» и «эффект времени».

Содержательно набор объединяет три взаимодополняющих класса источников.

Первый - официальная социально-экономическая и демографическая статистика, описывающая условия жизни сельских сообществ: численность и структура населения, рождаемость и миграционные притоки, жилищный фонд, доступ к централизованной питьевой воде и сетевому интернету, кадры социальной сферы (медицинские работники, преподаватели), а также показатели хозяйственной активности - поголовье крупного рогатого скота, площади пашен, наличие торговых объектов.

Второй класс - природно-ресурсные и экологические характеристики: площади категорий землепользования и земельного покрова (LULC - Land Use / Land Cover, землепользование и земельный покров) - пашни, пастбища, леса,

водные объекты, застроенные объекты и территории. Этот контур опирается на признанные в дистанционном зондировании подходы к классификации земной поверхности (семейство алгоритмов MODIS MCD12 и высокодетальные глобальные карты с разрешением 30 м) [43,44].

Третий класс - мультиспектральные спутниковые индексы, каждый из которых имеет ясный физический смысл: NDVI отражает фотосинтетическую активность растительности [45,46]; EVI - его уточнённая версия, более устойчивая при плотной растительности и в условиях дымки [47]; NDWI (Normalized Difference Water Index - нормализованный разностный водный индекс) и его модификация MNDWI (Modified NDWI - модифицированный нормализованный разностный водный индекс) выделяют открытые водные поверхности [48,49]; NDBI (Normalized Difference Built-up Index - нормализованный разностный индекс застройки) - классический индекс застройки [50]. Влагоёмкость и текстурные свойства поверхности оцениваются комбинациями ближнего и среднего инфракрасных (ИК) диапазонов (NDMI - Normalized Difference Moisture Index, нормализованный разностный индекс влажности, и NDTI - Normalized Difference Tillage Index, нормализованный разностный индекс обработки почвы; в составе производных признаков) [48,49]. Дополнительно используются показатели ночного света (NTL - Night-Time Lights, данные ночного освещения) - косвенный, но надёжный индикатор экономической активности и урбанизации, интерпретируемый через связь светового потока с плотностью населения и энергопотреблением [51].

В совокупности статистика, LULC и спектральные индексы образуют «мультидоменный» портрет каждого округа, отражающий сразу экономическое, социальное и экологическое измерения устойчивого развития. Ценность такой архитектуры двоякая. Во-первых, спутниковые признаки физически интерпретируемы (NDVI/EVI - «зелёная масса», NDWI/MNDWI - вода, NDBI - интенсивность застройки), глобально воспроизводимы и сопоставимы во времени [45–50]. Во-вторых, социально-экономические показатели со-

поставимы между округами благодаря унифицированной статистической отчётности. Категории LULC связывают эти два мира: они переводят «площадь» в «функцию», позволяя соотносить биофизическую основу сельского хозяйства (пашни, пастбища) с инфраструктурными и социальными индикаторами и отслеживать потенциал и ограничения природного капитала (леса, водные объекты) [43,44].

Для практического использования массива важна прослеживаемость: каждый показатель должен иметь понятный источник, единицы измерения и пространственно-временное разрешение. Межведомственную и межпроектную совместимость обеспечивают формальные метаданные по стандарту ISO 19115 (качество, происхождение, пространственная привязка, ограничения использования) и публикация производных геослоёв через стандартизованные OGC-сервисы: WMS - для растровых представлений, WFS - для векторных атрибутов [50,52]. Такая организационная дисциплина упрощает верификацию и повторное использование данных, их стыковку с кадастром, границами административных единиц и гидрографией, а главное - делает прозрачными все последующие аналитические процедуры.

Наконец, сильная сторона массива - баланс «широты охвата» и «качества измерения». Статистика даёт прямые измерения социально-экономического состояния, а спектральные индексы и LULC - физически обоснованные косвенные показатели, улавливающие пространственную неоднородность природных условий и изменений землепользования. Вместе они образуют насыщенное признаковое пространство (46 количественных переменных на запись), достаточное и для содержательного сравнения округов, и для анализа межгодовой динамики, и для последующего применения объяснимых моделей - без потери воспроизводимости и сопоставимости во времени.

Программная реализация всех описанных далее расчётных процедур - нормировки данных, корреляционного анализа, расчёта интегрального индекса, DEA-моделей и анализа факторов-драйверов методами машинного обучения - приведена в

Приложении 2 в виде полного воспроизводимого скрипта на языке Python.

**3.2. Предварительная обработка данных:
корреляционный анализ**

Прежде чем искать взаимосвязи между показателями, данные необходимо привести к сопоставимому виду: социальные, экономические и спутниковые переменные измеряются в несравнимых единицах. Поэтому каждый числовой показатель был нормализован - приведён к интервалу [0-1] методом Min-Max. Это выравнивает масштабы, улучшает численную устойчивость расчётов и делает сравнение коэффициентов корреляции корректным для разнотипных показателей [53,54]. Как нормализация меняет форму распределений, показано на рисунке 3.2.1 а–г.

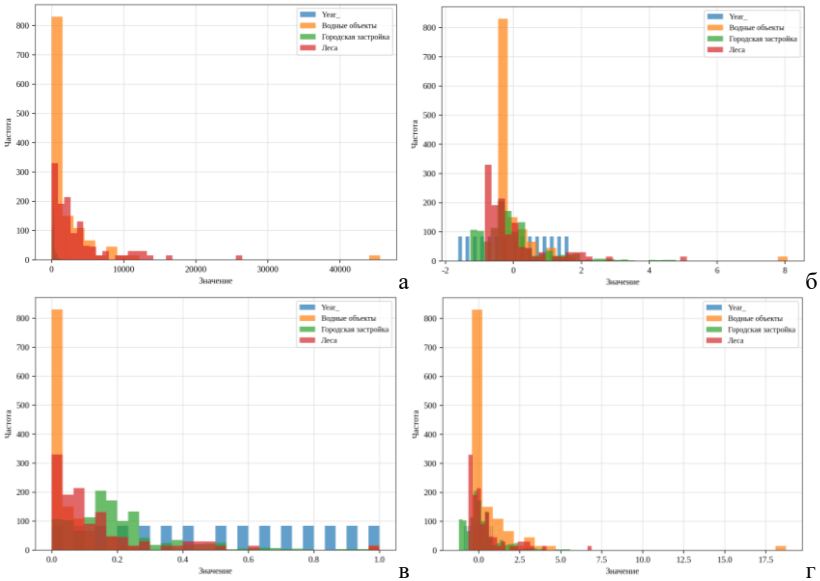


Рисунок 3.2.1 – Сравнение распределений до и после нормализации (а – оригинальные данные, б – StandardScaler, в – MinMaxScaler, г – RobustScaler)

Базовой мерой взаимосвязи выбран коэффициент линейной корреляции Пирсона r - классический показатель того, насколько согласованно изменяются две величины [47,54,55]. Для его интерпретации использованы общепринятые пороги: связь «очень сильная» при $|r|>0,90$, «сильная» при $0,70\leq|r|\leq0,90$, «средняя» при $0,50\leq|r|<0,70$ и «слабая» при $|r|<0,50$ [7,8]. Поскольку коэффициент Пирсона улавливает только линейные зависимости и чувствителен к выбросам, для проверки устойчивости выводов предусмотрены его ранговые аналоги - коэффициенты Спирмена и Кендалла, сравнивающие не сами значения, а их порядок [48,56–59]. Применение ранговой корреляции Спирмена для выявления трендов в данных мониторинга окружающей среды подробно рассмотрено в [60].

Расчёт выполнен на полной панели «округ–год»: 1260 наблюдений и 46 числовых переменных, что даёт 1035 возможных пар показателей. По итогам расчётов выявлено 33 пары с «сильной» связью ($|r|>0,70$).

Результаты представлены в виде нескольких графиков:

- корреляционная матрица нормализованных переменных (рисунок 3.2.2);

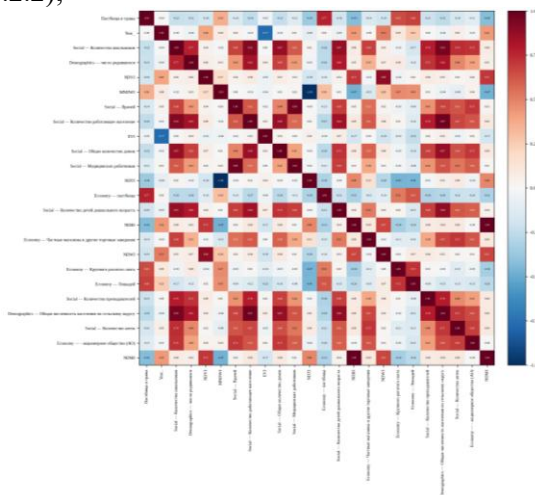


Рисунок 3.2.2 – Корреляционная матрица нормализованных переменных с сильными связями ($|r|>0.7$)

- матрица с цветовым кодированием силы связи (рисунок 3.2.3);

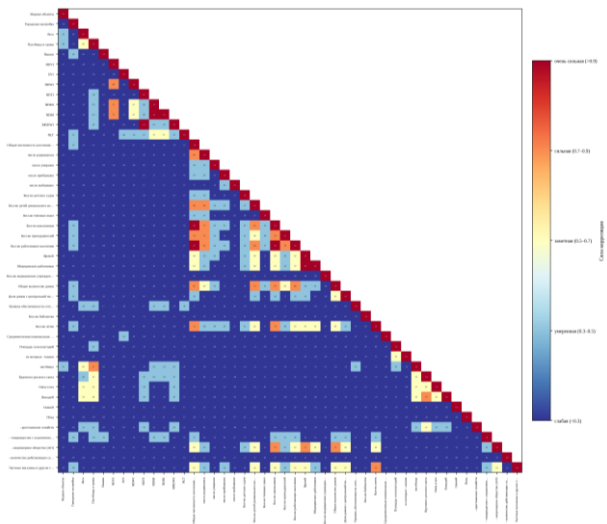


Рисунок 3.2.3 – Корреляционная матрица с цветовым кодированием по силе связи

- матрица средних корреляций между категориями переменных (рисунок 3.2.4).

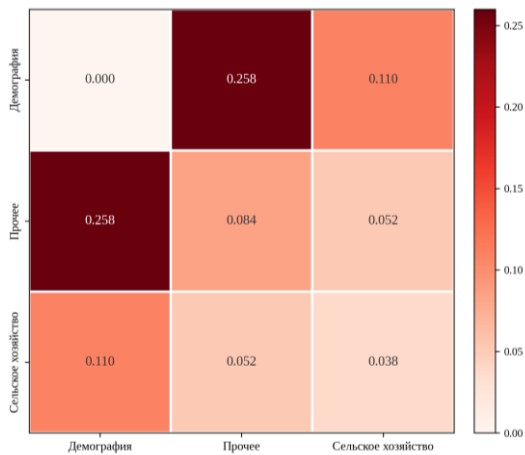


Рисунок 3.2.4 – Средние корреляции между категориями переменных

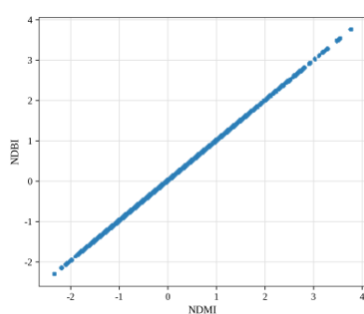
Такой результат закономерен для данных о сельских территориях: многие показатели растут согласованно (например, демографические и социальные индикаторы), а спутниковые индексы, вычисляемые из близких спектральных каналов, неизбежно образуют коррелирующие кластеры.

В структуре сильных связей выделяются три содержательных блока.

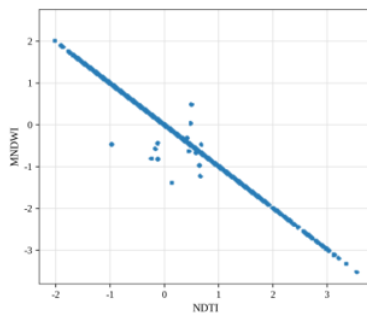
1. Первый блок - демографический и социальный. Рост общей численности населения сопровождается увеличением числа школьников и доли занятого населения; согласованно движутся и инфраструктурные показатели - жилищный фонд, обеспеченность услугами, кадры социальной сферы. Самые высокие значения зафиксированы для пар «население ↔ школьники» (примерно $r \approx 0,95$) и «врачи ↔ медработники» (около $r \approx 0,96$). За этим стоит масштабный эффект: чем крупнее и плотнее поселение, тем выше нагрузка на социальные институты и жилищную инфраструктуру. Отдельные парные связи показаны на рисунке 3.2.5 а,б,в,г,д,е.

2. Второй блок - производственно-аграрный. поголовье крупного рогатого скота, площади пашен и пастбищ и связанные с ними хозяйственные переменные образуют согласованные группы. Эти корреляции отражают и агроклиматическую основу (земельный ресурс определяет производственный потенциал), и технологические особенности управления ресурсами на уровне округа.

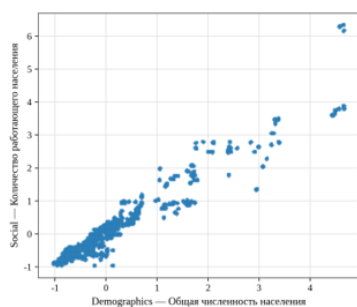
3. Третий блок - спектрально-экологический (спутниковые индексы и LULC). Индексы растительности (NDVI/EVI), воды (NDWI/MNDWI), застройки (NDBI) и влагочувствительные показатели (NDMI/NDTI) демонстрируют ожидаемо высокую сопряженность. Показательны почти идеальная положительная зависимость NDMI–NDBI ($r \rightarrow 1,00$) и очень сильная отрицательная NDTI–MNDWI ($r \rightarrow -1,00$). Причина - в самой физике измерений: индексы строятся из одних и тех же спектральных каналов ближнего и среднего ИК-диапазонов, а типы земельного покрова (открытая вода, наращивание урбанизированной площади) «тянут» их в противоположные стороны [47,48], [41,49–51].



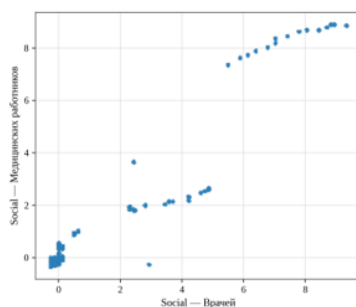
а



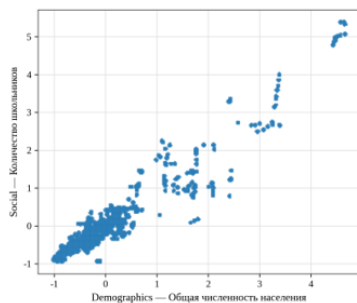
б



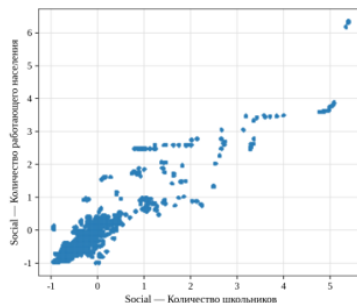
в



г



д



е

Рисунок 3.2.5 – Scatter plots для нормализованных переменных с сильными корреляциями (а – корреляция между NDBI и NDMI, б – корреляция между MNDWI и NDTI, в – количество работающего населения и общая численность населения по сельскому округу, г – медицинские работники и врачи, д – количество школьников и общая численность населения, е – количество работающего населения и школьников)

Иллюстрация топ-10 пар с самыми высокими значениями приведена на рисунке 3.2.6.

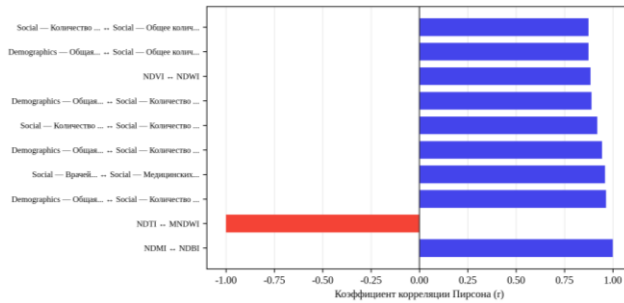


Рисунок 3.2.6 – Топ-10 пар с сильной корреляцией ($|r| > 0.7$)

Из этих наблюдений следуют два практических вывода. Первый: кластеры с очень высокими $|r|$ означают избыточность информации (мультиколлинеарность). Модель, получившая несколько почти одинаковых предикторов, рискует «посчитать дважды» одну и ту же вариативность. Поэтому нужны осознанные решения: удалить дубли, объединить близкие показатели в тематические композиты (например, единый индикатор «жилищный фонд») либо применять методы, устойчивые к мультиколлинеарности. Характерные распределения и выбросы корреляций иллюстрируют рисунки 3.2.7–3.2.8.

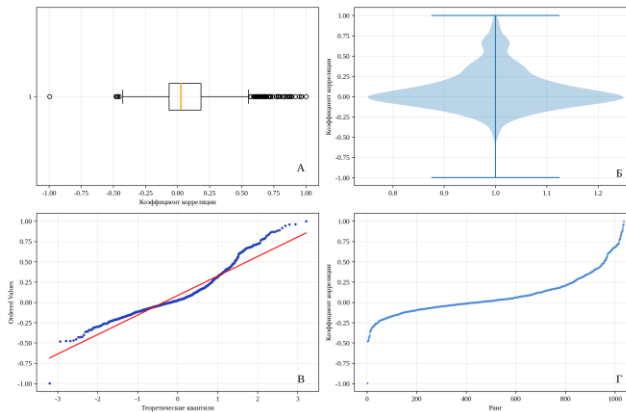


Рисунок 3.2.7 – Анализ выбросов и распределения корреляций (А – box plot, Б – Violin, В – Q-Q, Г – корреляции vs ранги)

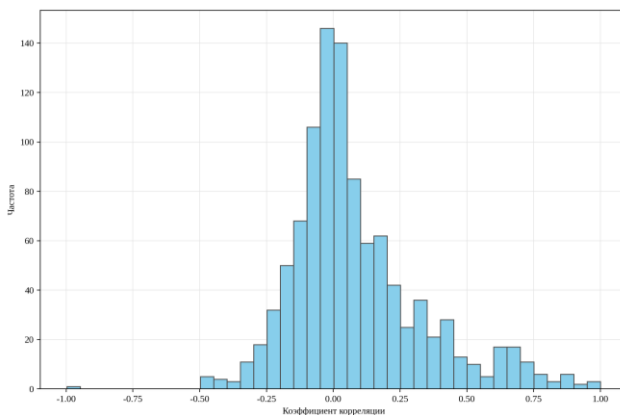


Рисунок 3.2.8 – Распределение коэффициентов корреляции после нормализации

Второй вывод: содержательная интерпретируемость кластеров упрощает все последующие шаги. Спектральные индексы имеют физическую трактовку (биомасса, вода, застройка), социально-демографические показатели - институциональную и поведенческую (школы, здравоохранение, жильё), аграрные - ресурсную и технологическую (земли, поголовье). Это даёт понятную основу для формализации признаков при дальнейшем анализе.

Методологическая оговорка: при одновременной проверке тысячи пар корреляций часть «значимых» связей неизбежно оказывается случайной — это проблема множественных сравнений. При необходимости строгого контроля применяется процедура Benjamini–Hochberg, ограничивающая долю ложных открытий (FDR, False Discovery Rate) и корректирующая интерпретацию p -значений без чрезмерно консервативных поправок [61]. Для прикладной задачи отбора признаков по сельским территориям, как правило, достаточно использованного здесь порогового правила по $|r|$ в сочетании с содержательной интерпретацией и последующей проверкой устойчивости в моделях. Влияние нормализации на корреляции иллюстрирует рисунок 3.2.9 а, б.

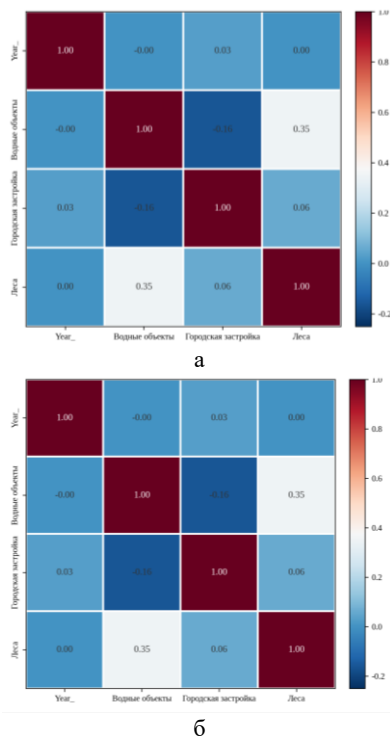


Рисунок 3.2.9 – Корреляции на оригинальных и нормализованных данных (а – корреляция на оригинальных данных, б – корреляция на нормализованных данных)

3.3 Оценка агропроизводственной эффективности сельских округов методом Data Envelopment Analysis

Для интегральной оценки эффективности округов использован непараметрический метод анализа оболочки данных (Data Envelopment Analysis, DEA) [62–64]. Его суть можно описать просто: каждый округ рассматривается как «единица принятия решений» (DMU, Decision Making Unit), которая расходует ресурсы (входы) и производит результаты (выходы). Метод находит наилучшие фактически наблюдаемые сочетания «ресурсы → результаты» и строит по ним границу эффективности - эмпири-

ческую «оболочку» Парето-оптимальных комбинаций. Эффективность каждого округа измеряется его положением относительно этой границы. Принципиальное преимущество DEA в том, что он не требует заранее задавать вид производственной зависимости и корректно сравнивает территории сразу по нескольким входам и выходам.

Расчёт выполняется отдельно для каждого года наблюдения: каждый округ (DMU) описывается своим набором входов и выходов. Использована базовая модель CCR (Charnes–Cooper–Rhodes), предполагающая постоянную отдачу от масштаба (CRS), во входной ориентации (input-oriented): такая постановка отвечает на вопрос, насколько можно сократить ресурсы, не теряя достигнутых результатов [62]. Формально задача для оцениваемого округа записывается как линейная программа (формула 1):

$$\min_{\theta, \lambda, s^-, s^+} \theta \text{ s.t. } X\lambda + s^- \leq \theta x_0, Y\lambda - s^+ \geq y_0, \lambda \geq 0, s^- \geq 0, s^+ \geq 0, \quad (1)$$

где X и Y - матрицы входов и выходов по всем DMU; λ - вектор весов, задающий «эталонную комбинацию» лучших округов, с которой сравнивается оцениваемый; s^- и s^+ - слэки, то есть остаточные резервы (избыточность конкретных входов и недостаток конкретных выходов); $\theta \in [0, 1]$ - радиальная мера технической эффективности: чем ближе к 1, тем эффективнее округ [65, 66].

Той же задаче соответствует эквивалентная по оптимальному значению θ дуальная (мультипликативная) запись: она максимизирует взвешенную сумму выходов при нормирующем ограничении на взвешенную сумму входов (формула 2):

$$\max_{u, v} u^T y_0 \text{ s.t. } v^T x_0 = 1, u^T y_j - v^T x_j \leq 0 \forall j, u \geq \varepsilon, v \geq \varepsilon, \quad (2)$$

где u и v - неотрицательные веса выходов и входов, подбираемые самой моделью наиболее выгодным для оцениваемого округа образом; $\varepsilon > 0$ - малый регулятор, исключающий вырожденные решения [62–64]. Если есть основания ожидать переменную отдачу от масштаба (VRS), модель расширяется до BCC (Banker–

Charnes–Cooper) добавлением одного ограничения [67]; в настоящей работе для сопоставимости округов принята CRS-логика модели CCR.

Выбор входов и выходов. В расчётах использованы 3 входа и 4 выхода, охватывающие ресурсный контур, инфраструктурно-социальные результаты и биофизическую «прокси»-составляющую:

- Входы (ресурсы):
 - a) *Пастбища и травы* (площадь),
 - b) *Пашни* (площадь),
 - c) *Численность работающего населения* (трудовой ресурс).
- Выходы (результаты):
 - d) *Доля домов с центральной питьевой водой*, % (инженерная инфраструктура),
 - e) *Уровень обеспеченности сетевым интернетом*, % (цифровая доступность),
 - f) *Число прибывших* (миграционная привлекательность/соц. динамика),
 - g) *(NDVI (спутниковый индекс растительности как прокси биопродуктивности) [68–71].*

h) Обоснование выбора. Входы отражают ключевые ресурсы аграрной системы: землю (пастбища и пашни как основные категории сельскохозяйственных угодий) и труд (численность работающего населения) - все они управляемы на уровне округа, что и требуется для input-ориентированной модели. Выходы намеренно охватывают не только производственный результат: доля домов с питьевой водой и обеспеченность интернетом фиксируют инфраструктурный результат использования ресурсов; число прибывших служит поведенческим индикатором привлекательности округа (люди «голосуют ногами» за качество среды); NDVI - независимый от ведомственной отчётности спутниковый прокси биопродуктивности угодий. Такой состав снижает риск «производственного перекоса», когда эффективными признаются округа с высокой отдачей земли, но деградирующей социальной средой.

Логика этого набора такова: округ эффективен, если при минимальных площадях и трудовых затратах он обеспечивает высокий инфраструктурно-социальный результат и агробиоло-

гический потенциал. Основные распределения DEA-индекса приведены на рисунке 3.3.1 а,б,в,г,д,е. Соотношение числа округов (DMU) и числа показателей удовлетворяет «правилам разумного выбора» для DEA [72], что снижает риск искусственного попадания слишком многих округов на границу эффективности. Программная реализация DEA-модели (входо-ориентированная CCR, супер-эффективность, индекс Мальмквиста) приведена в Приложении 2 (модуль `dea_lp.py`).

Как читать результаты. Линейная программа решается для каждого округа отдельно. Оптимальное значение θ^* показывает, насколько можно пропорционально сократить все входы, чтобы выйти на границу эффективности при сохранении результатов: например, $\theta^*=0,7$ означает резерв экономии ресурсов около 30%. Ненулевые компоненты λ указывают «пиры» - реальные эффективные округа, комбинация которых служит образцом для отстающего. Ненулевые слэки s^- , s^+ выявляют адресные резервы (избыток конкретного ресурса или недобор конкретного результата), не устранимые одним пропорциональным сокращением; их учёт критичен для корректной постановки целей улучшения [63,64,73]. Итог расчёта – распределение θ^* на отрезке $[0,1]$, где $\theta^*=1$ означает принадлежность к границе эффективности (при нулевых слэках - строгую эффективность).

Результаты по массиву. Распределение получилось типичным для «реального сектора»: значительная доля округов с невысокой эффективностью и узкая группа эталонов на границе. Средняя эффективность составила около 0,456, медиана - ~0,355, минимум - ~0,033, максимум - 1,000. По категориям: низкая эффективность ($<0,5$) - около 65% округов, средняя (0,5–0,8) - ~17%, высокая (0,8–1,0) - ~15%, максимальная ($=1,0$) - ~3%. Округа последней группы выступают «пирами» для более слабых: именно на их комбинации ориентируются планы улучшений. Практический смысл: для большинства округов существуют достижимые - подтверждённые практикой лучших территорий - цели по снижению ресурсной нагрузки при сохранении или наращивании результатов. Распределение DEA-индекса по регионам и его динамика по годам представлены на рисунке 3.3.2 а,б. В приложении 2 приведены

карты распределения индекса DEA по округам за 15 лет (с 2010 по 2024 годы).



Рисунок 3.3.1 – Анализ DEA индекса эффективности (а – распределение, б – box-plot, в – динамика по годам, г – топ-20 округов, д – DEA индекс по регионам, е – распределение по категориям)

Интерпретация входов/выходов и целеполагание:

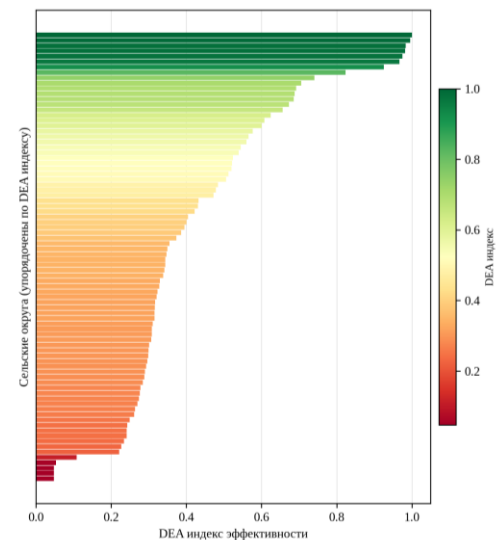
- Ресурсные площади (*пастбища, пашины*) и трудовой компонент отражают объём «затрат» в аграрной системе; минимизация их при данных результатах - естественная цель input-ориентированного DEA;

- Выходы, представленные инженерной и цифровой обеспеченностью, миграционной динамикой и NDVI, фокусируют внимание не только на непосредственной агропродуктивности, но и на качестве жизни и доступности услуг, что согласуется с повесткой устойчивого сельского развития [68–71];

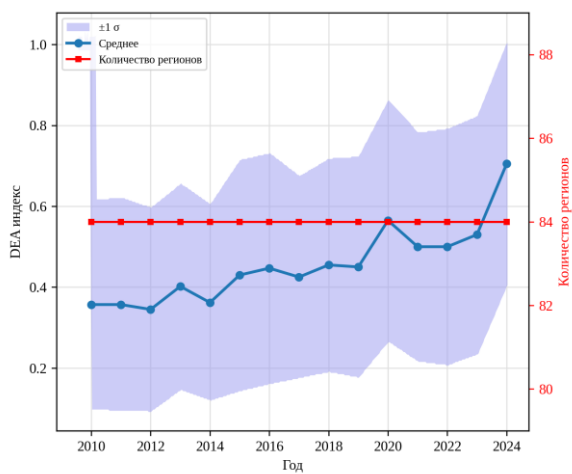
- Целевые значения для неэффективных DMU выводятся из оптимального решения $(\theta^*, \lambda^*, s^{-*}, s^{+*})$: радикально – пропорциональным сокращением входов (θ^*) , и дополнительно - адресным устранением слэков (например, «сжать» избыток одной из площадей или повысить конкретный выход);

- С учётом гетерогенности территорий, для управленческих целей полезны пир-группы (референсные «команды» эффективных округов), агрегируемые по типологии LULC, климату и инфраструктурному профилю.

Надёжность и расширения. Слабое место модели CCR – чувствительность к единичным выбросам, способным «поднять» границу эффективности. Для диагностики устойчивости применяются супер-эффективность Andersen–Petersen [74], кросс-эффективность [75], бутстреп-оценки доверительных интервалов [76] и устойчивые к шуму варианты DEA [46]. Если ожидается переменная отдача от масштаба, полезно сравнить CCR и ВСС: расхождение оценок покажет вклад масштабного эффекта [67]. Для панельных данных применим индекс Мальмквиста, раскладывающий динамику эффективности во времени на технологический прогресс и собственно изменение эффективности [77]. Взгляд на DEA как на непараметрическую регрессию, позволяющий отделять статистический шум от собственно неэффективности, развит в [78].



а



б

Рисунок 3.3.2 – График распределения DEA индекса по округам
(а – распределение по регионам, б – динамика по годам)

Итог раздела: DEA дал прозрачную и интерпретируемую метрику относительной эффективности округов. Она порождает конкретные целевые ориентиры по входам и выходам, выявляет эталонные практики и создаёт основу для последующих блоков работы - пространственного анализа зон отставания и лидерства, а также объяснимых моделей, устанавливающих драйверы различий в θ^* .

Глава 4. МАШИННОЕ ОБУЧЕНИЕ ДЛЯ ОБЪЯСНЕНИЯ И ПРОГНОЗИРОВАНИЯ ИНДЕКСА DATA ENVELOPMENT ANALYSIS

4.1. Random Forest

Случайный лес (Random Forest) — это ансамбль из множества деревьев решений: каждое дерево обучается на случайной бутстрепп-подвыборке данных и случайном подмножестве признаков, а итоговый прогноз усредняется [79]. Для гетерогенных табличных данных, подобных нашим, такая конструкция подходит по двум причинам: усреднение по деревьям снижает разброс предсказаний, а случайный отбор признаков не позволяет нескольким сильным предикторам «монополизировать» модель. При этом метод не требует предположений о линейности или монотонности - он сам улавливает широкий спектр нелинейностей и взаимодействий. На нашем массиве лес показал наивысшую долю объяснённой вариации при устойчивости между фолдами кросс-валидации, что согласуется с обширной эмпирикой по табличным наборам, где RF (Random Forest) часто даёт оптимальный компромисс «точность/робастность» [79-78]. Классическая оценка «важности» признаков в деревьях основана на том, насколько признак снижает неопределённость при разбиениях, однако у неё есть известные недостатки [80]:

- признаки с большим числом возможных разбиений часто выглядят «важнее», чем они есть на самом деле;
- при сильной корреляции (коллинеарности) один признак может «перетягивать» важность у другого.

Поэтому для практического, особенно управленческого, использования предпочтительнее более надёжные методы:

- пермутационная важность: признак случайно перемешивается, и измеряется, насколько от этого ухудшается качество модели;
- апостериорные (post-hoc) объяснения: SHAP (SHapley Additive exPlanations - аддитивные объяснения Шепли), раскладывающий каждый прогноз на вклады отдельных признаков (рисунок 4.1.1), и графики частичных зависимостей (PDP - Partial

Dependence Plot, ALE - Accumulated Local Effects), показывающие характер влияния признака [81–83].

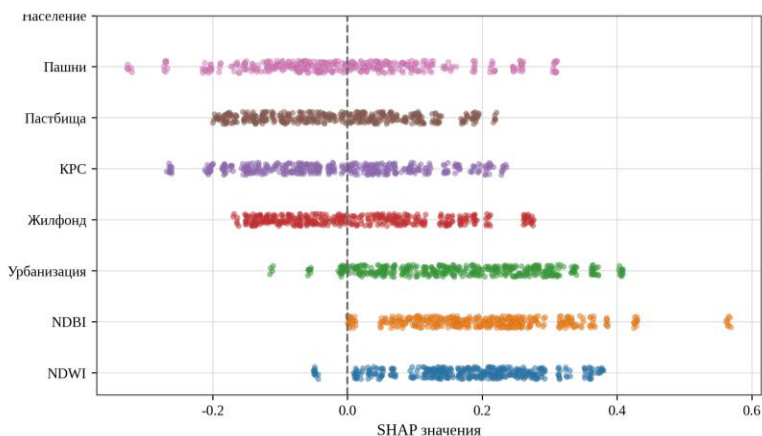


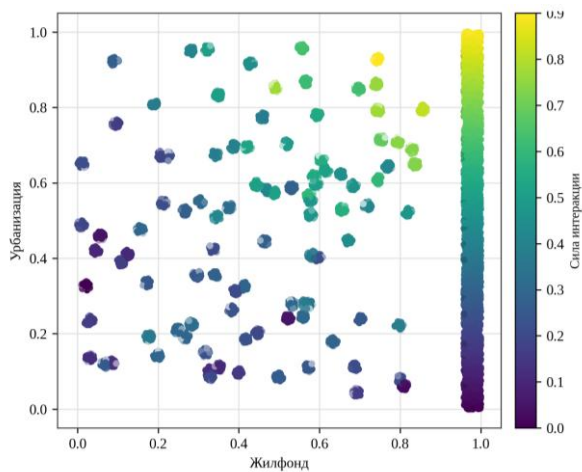
Рисунок 4.1.1 – SHAP Beeswarm диаграмма для модели Random Forest

По итогам анализа в верхних строках рейтинга факторов устойчиво оказываются: производственный контур животноводства (крупный рогатый скот) как индикатор аграрной мощности; природно-ресурсная компонента «Леса», регулирующая водный режим и биопродуктивность; жилищно-социальные индикаторы (общее количество домов, количество преподавателей) - инфраструктурное ядро качества жизни; водный спектральный индекс NDWI как прокси доступности влаги; а также индикаторы урбанизации (в том числе NDBI). Локальные объяснения выявляют пороговость и зоны насыщения: у NDWI существует диапазон «полезной влаги» с максимальной отдачей, за пределами которого эффект ослабевает; вклад урбанизации в сочетании с жилищным фондом положителен до порога насыщения сетевой инфраструктуры, после чего прирост эффективности замедляется. Эти эффекты подтверждаются ALE-кривыми и ICE-профилями (Individual Conditional Expectation) и формируют наглядные «окна эффективности» для управленческих сценариев. В итоге Random

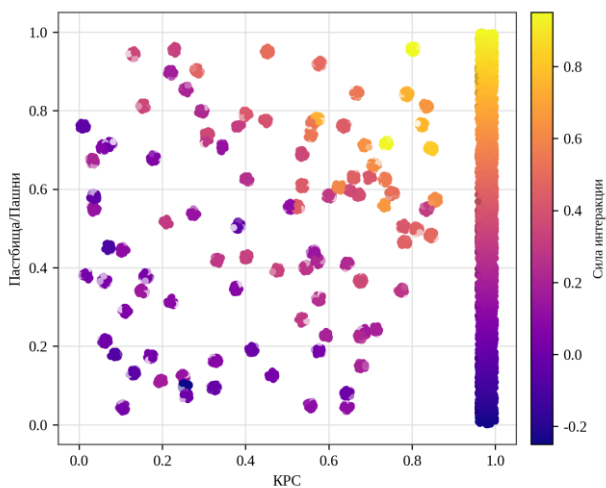
Forest закрепляется как базовый производственный инструмент: высокая точность, устойчивость к мультиколлинеарности и полноценная объяснимость через современный набор post-hoc методик [80,82].

4.2. Градиентный бустинг деревьев (последовательная аппроксимация сложных нелинейностей)

Градиентный бустинг устроен иначе, чем случайный лес: деревья строятся не независимо, а последовательно - каждое следующее приближает антиградиент функции потерь, то есть исправляет ошибки, накопленные предыдущей композицией [84]. Такая пошаговая схема тонко воспроизводит пороговые структуры, насыщения и взаимодействия высокого порядка, которые в классической регрессии потребовали бы трудоёмкой ручной инженерии признаков [81,82,84]. На нашем наборе бустинг лишь умеренно уступил RF по точности, но нередко превосходил его в чёткости локальных зависимостей: по NDWI и NDBI выделяются участки с максимальной маржинальной отдачей, что удобно для нормирования целевых порогов. Устойчивость обеспечивается регуляризацией - малым шагом обучения, ограничением глубины деревьев, стохастическими подвыборками и ранней остановкой по валидации [84]; в такой конфигурации качество согласовано между фолдами, а поведение модели на периферии признакового пространства, где наблюдений мало, остаётся предсказуемым [85]. Как и лес, бустинг объясняется через SHAP/PDP/ALE, но его профили чаще «ступенчатые» (рисунок 4.2.1) - и именно это удобно для формулирования практических правил: например, определить диапазон NDWI, в котором инвестиции в базовую инфраструктуру дают наивысшую отдачу, или уровень урбанизации, после которого без синхронного наращивания жилищно-инженерных мощностей эффективность растёт заметно медленнее [83]. Иными словами, бустинг не просто ранжирует факторы - он подсказывает рабочие диапазоны их значений.



а



б

Рисунок 4.2.1 – Матрица SHAP-интеракций факторов
(а – SHAP-интеракция «Жилфонд × Урбанизация»,
б – SHAP-интеракция «КРС × Пастбища/Пашни»)

4.3. Регрессия на опорных векторах (SVR с RBF-ядром как гладкая регуляризованная аппроксимация)

Метод опорных векторов для регрессии (SVR, Support Vector Regression) решает задачу иначе, чем деревья: он строит максимально гладкую функцию, штрафую лишь ошибки, выходящие за допустимый «коридор» шириной ϵ . Нелинейность вводится через ядро - в нашем случае RBF (Radial Basis Function, радиальная базисная функция), задающее гладкую карту признакового пространства. Для табличных данных умеренного объёма SVR нередко оказывается оптимальной «гладкой» альтернативой деревьям, особенно когда нужна устойчивая аппроксимация в областях редких комбинаций признаков [86]. На нашем наборе SVR уступил ансамблям по точности, но давал более стабильные предсказания там, где деревья демонстрировали высокую дисперсию локальных оценок. Для метода принципиальны стандартизация признаков и аккуратный подбор параметров (C , γ , ϵ) по кросс-валидации: чрезмерно большие γ и C ведут к переобучению, слишком малые - к «переглаживанию» [79,86]. Объяснимость достигается методами Kernel/Permutation-SHAP и ALE-кривыми (рисунок 4.3.1 а,б).

Например, вклад NDWI существенно зависит от уровня жилищно-социальной обеспеченности - при среднем и высоком жилищном фонде маргинальная отдача влаги на эффективность заметно выше, чем при низком; выраженные комбинационные эффекты демонстрируют и «Леса» с «КРС» (крупный рогатый скот) в связке с пашнями и пастбищами, отражая природно-ресурсную структуру округа. Роль SVR в общей картине - «стабилизатор»: рекордной точности он не даёт, но укрепляет доверие к профилям маргинальных эффектов в тонких зонах признакового пространства и полезен для стресс-тестов сценариев.

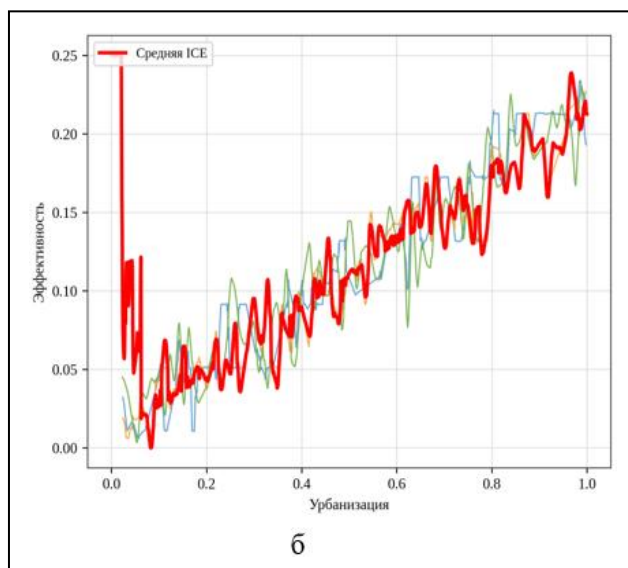
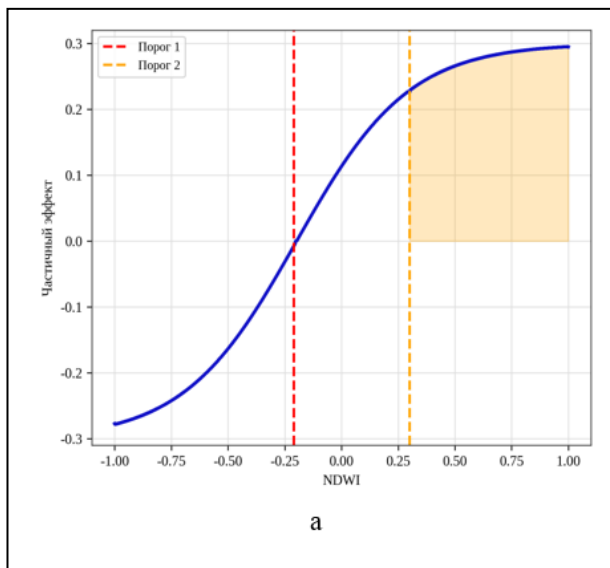


Рисунок 4.3.1 – Интерпретация модели: PDP/ALE профиль для NDWI и ICE-кривые для Урбанизации (а – PDP/ALE профиль для NDWI, б – ICE-кривые «Урбанизация → Эффективность»)

4.4. Полносвязные нейронные сети (гибкость аппроксимации при повышенных требованиях к валидации)

Полносвязные нейронные сети - универсальные аппроксиматоры: теоретически они способны воспроизвести любую непрерывную зависимость. Однако на табличных данных малого и среднего объёма их преимущество перед деревьями не гарантировано: сети чувствительнее к масштабированию, инициализации, режиму обучения и регуляризации и дают больший разброс результатов [87]. В наших условиях (стандартизация признаков, несколько умеренно-широких скрытых слоёв, batch normalization, dropout) нейросеть уступила ансамблям по точности и стабильности, но внесла свой вклад - «мягкие» нелинейности: профили SHAP и PDP по NDWI и NDBI получались более плавными, как и взаимодействия «урбанизация × жилищный фонд» и «КРС × пастбища/пашни». Чтобы объяснениям можно было доверять, мы усредняли их по нескольким запускам с фиксированными сид-значениями - это заметно снижало дисперсию локальных вкладов. Практический вывод: нейросеть уместна как вспомогательный сценарный инструмент там, где деревья дают избыточно «ломанные» профили, а требуется гладкая реакция на малые изменения признаков (например, при поэтапном улучшении цифровой доступности или распределённых мерах по водообеспечению). Когда же приоритетны воспроизводимость и прозрачность управленческих рекомендаций, предпочтительны ансамбли деревьев.

4.5. Elastic Net и сводная аналитика результатов

Elastic Net - линейная регрессия с двойной регуляризацией: L1-штраф обнуляет коэффициенты малозначимых признаков, выполняя их отбор, а L2-штраф стабилизирует оценки при мультиколлинеарности [88]. В нашей задаче она ожидаемо уступает нелинейным моделям - зависимость индекса эффективности насыщена порогами, насыщениями и взаимодействиями, - однако решает три полезные задачи:

- проверка «здравого смысла» - знаков и масштабов эффектов на стандартизованной матрице [81,82];

– компактная спецификация модели для коммуникации с управленцами;

– минимальный набор индикаторов для лёгкой отчётности.

На фоне остальных методов Elastic Net играет роль «коммуникационного моста»: он переводит сложную нелинейную структуру в краткие воспроизводимые линейные ориентиры.

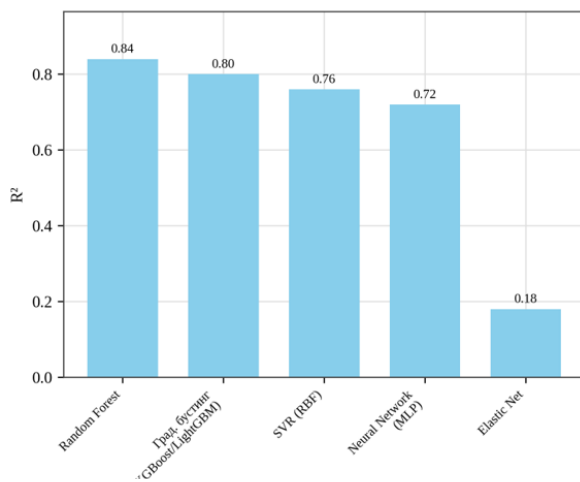
Отдельно зафиксируем протокол валидации, единый для всех моделей ML (Machine Learning - машинное обучение) и DL (Deep Learning - глубокое обучение). Данные делились на обучающую и тестовую выборки в соотношении 75/25 с фиксированным начальным состоянием генератора случайных чисел (random_state = 42), что обеспечивает воспроизводимость; устойчивость оценок дополнительно контролировалась 5-кратной кросс-валидацией. Поскольку наблюдения образуют панель «округ–год», для исключения утечки информации между связанными наблюдениями применялась группировка по округам (GroupKFold), а для рекуррентных моделей - разбиение по времени: ранние годы в обучение, поздние в тест (подробнее см. раздел 5.4). Метрики качества далее приводятся для тестовой выборки и кросс-валидации в рамках этого протокола.

Сводное сравнение качества и интерпретируемости моделей (целевая переменная - DEA-индекс округа) фиксирует следующую картину (таблица 4.5.1); сравнительный анализ точности приведён на рисунках 4.5.1 и 4.5.2.

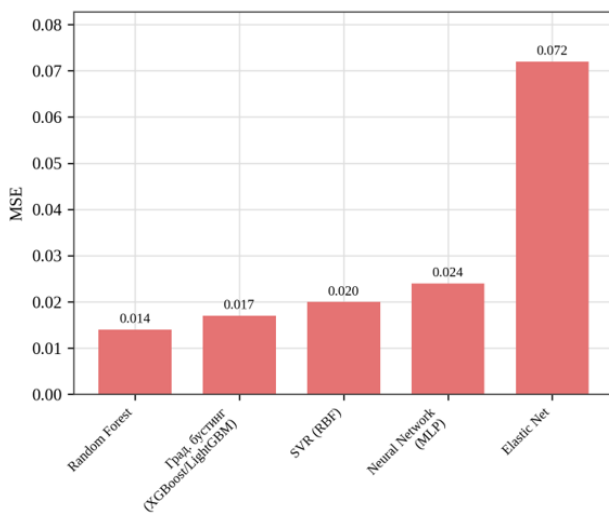
Таблица 4.5.1

**Сравнение качества и интерпретируемости моделей
машинного обучения**

Модель	R ² (test)	R ² (5- fold CV)	MSE (test)	MAE (test)	Интерпрети- руемость	Чувств-ть к масштабированию	Комментарий по рискам
Random Forest	0.84	0.80	0.014	0.073	Пост- <u>hoc</u> (SHAP/ALE)	Низкая	Надёжная, базовый эталон
Градиентный <u>бустинг</u> (XGBoost/LightGBM)	0.80	0.76	0.017	0.091	Пост- <u>hoc</u> (SHAP/ALE)	Низкая	Точен, <u>требов.</u> тюнинга
SVR (RBF)	0.76	0.72	0.020	0.105	Пост- <u>hoc</u> (SHAP/ALE)	Высокая	Гладкая аппроксимация
<u>Neural</u> Network (MLP)	0.72	0.50	0.024	0.105	Пост- <u>hoc</u> (SHAP/ALE)	Высокая	Нужна жёсткая валидация
<u>Elastic</u> Net	0.18	0.15	0.072	0.228	Прямая (β - <u>коэф.</u>)	Высокая	Базис, не для прогноза

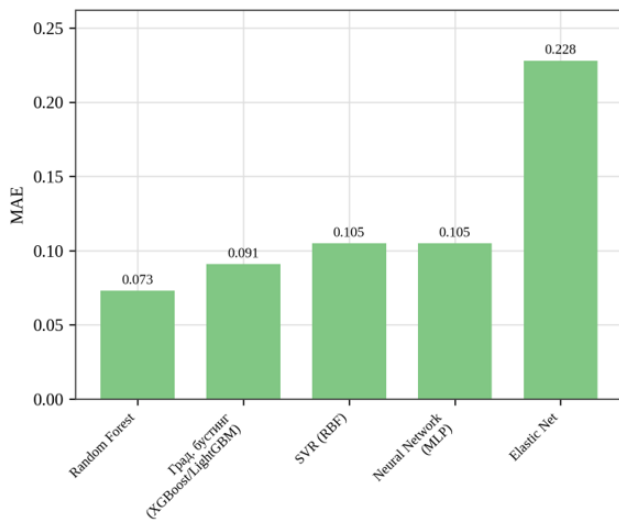


а

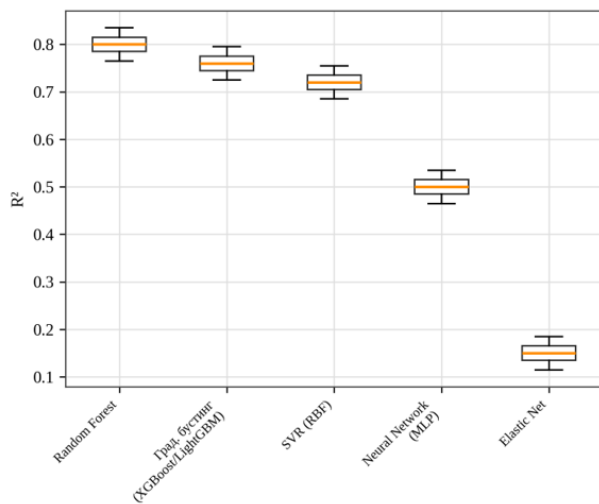


б

Рисунок 4.5.1 – Сравнение R^2 и MSE по моделям ML
(а – R^2 по моделям, б – MSE по моделям)



а



б

Рисунок 4.5.2 – Сравнение MAE и R^2 на кросс-валидации по моделям ML (а – MAE по моделям, б – R^2 по 5-fold CV)

По точности лидирует Random Forest (валидированный R^2 (коэффициент детерминации) порядка 0.84) с устойчивым поведением на кросс-валидации; градиентный бустинг близок к нему (R^2 около 0.80) и даёт особенно удобные для нормирования «рабочие диапазоны» ключевых факторов [84]; SVR обеспечивает сглаженную аппроксимацию (R^2 порядка 0.76) и полезен в редких конфигурациях признаков [86]; нейросеть достигает R^2 порядка 0.72 [87], добавляя «мягкие» нелинейности при большем разбросе; Elastic Net служит прозрачным линейным эталоном (R^2 около 0.18) [81,82], позволяющим обсуждать ядро сигналов простыми средствами. Важно, что все подходы сходятся в главном: набор ключевых драйверов один и тот же - водная обеспеченность (NDWI), жилищно-социальная инфраструктура (общее число домов, преподаватели), прокси урбанизации (в том числе NDBI), природно-ресурсный фактор «Леса» и производственный контур животноводства (KPC) (таблица 4.5.2).

Таблица 4.5.2

Ключевые факторы моделей машинного обучения и их эффекты

Фактор	Стабильность (% моделей)	Ожидаемый эффект (качественно)	Комментарий по взаимодействиям
Крупный рогатый скот (KPC)	80%	Сильная структурная роль производственного контура	Зависим от пашен/пастбищ
Леса	80%	Ресурсная/экологическая база, водный режим	В связке с NDWI и пашнями
Общее число домов	80%	Инфраструктурный каркас качества жизни	Синергия с урбанизацией
Число преподавателей	80%	Человеческий капитал и соц. инфраструктура	Связь с населением/жильём
NDWI	60%	Водная обеспеченность как прокси родуктивности	Пороговая зона «полезной влаги»
Урбанизация (в т.ч. NDBI)	60%	Положительно до порога насыщения сетей	Взаимосвязь с жильём
Ночное освещение (NTL/аналог)	40%	Прокси экономической активности	Взаимосвязь с урбанизацией
Пашни	40%	Ресурсная база растениеводства	С «KPC» и NDVI

Модели расходятся лишь в форме маржинальных эффектов и взаимодействий: ансамбли деревьев позволяют формулировать операциональные пороги и «окна эффективности», SVR и нейросеть (NN, Neural Network) подтверждают и сглаживают эти сюжеты, а Elastic Net конденсирует их в компактную спецификацию [79,81,82,84, 86,87].

Глава 5. ГЛУБОКОЕ ОБУЧЕНИЕ ДЛЯ ВЫЯВЛЕНИЯ ДРАЙВЕР-ФАКТОРОВ ИНДЕКСА DATA ENVELOPMENT ANALYSIS

5.1. «Компактные» и «глубокие» полносвязные сети в TensorFlow (SimpleTF / DeepTF)

Первое семейство DL-моделей - полносвязные нейронные сети (MLP, Multilayer Perceptron - многослойный перцептрон). На табличных данных они способны аппроксимировать зависимость практически любой формы между предикторами и целевой переменной - при условии аккуратной регуляризации, нормализации и валидации [87]. В нашем случае на вход сети подавались 33 признака (после исключения временной метки и переменных, участвовавших в формировании DEA), предварительно приведённые к стандартной шкале (z-нормировка): это критично для устойчивого обучения с адаптивными оптимизаторами типа Adam [89] и упрощает поиск минимума функции потерь. В качестве функции потерь использовалась среднеквадратичная ошибка:

$$L(\theta) = (1/n) \times \sum (\hat{y}_i - y_i)^2 + \lambda \|\theta\|_2^2 \quad (3)$$

где: $L(\theta)$ - функция потерь; n - количество наблюдений; $\hat{y}_i$ - предсказанное значение; y_i - фактическое значение; λ - коэффициент регуляризации; $\|\theta\|_2^2$ - L2-норма параметров.

Регуляризация включала L2-штраф (weight decay) и dropout - случайное «выключение» части нейронов при обучении [90]; в ряде конфигураций применялась batch normalization [91], стабилизирующая распределения активаций и ускоряющая сходимость. Сравнивались две конфигурации. «Компактная» сеть (SimpleTF) - один-два скрытых слоя умеренной ширины - показала наилучший баланс выразительной мощности и способности к обобщению: ранняя остановка по валидации (patience 10–20 эпох) фиксировала точку минимума обобщённой ошибки и не давала модели переобучиться на случайных флуктуациях. «Глубокая» вариация (DeepTF) с большим числом слоёв и нейронов была формально мощнее, но выигрыша в качестве

62

не дала, а разброс результатов между фолдами вырос. Это типичный эффект для табличных наборов среднего размера: избыточно «широкая» сеть перераспределяет вес между сильно коррелированными признаками и накапливает шум в весах [87].

Ценность DL в табличных задачах не сводится к точности (хотя «компактная» сеть оказалась одной из лучших) - она ещё и в возможности получать гладкие, согласованные объяснения драйверов. Мы системно применяли три взаимодополняющих метода атрибуции: пермутационную важность (устойчива к масштабированию, но чувствительна к коррелированным признакам), SHAP-подходы (Kernel или Deep - в зависимости от реализации) и интегральные градиенты (IG - Integrated Gradients, интегрированные градиенты) [83,92, 93]. Вместе они дают три уровня картины: глобальные ранги факторов, устойчивые при перезапусках; локальные профили влияния вблизи конкретных наблюдений; агрегированные PDP/ALE-профили с порогами и зонами насыщения. Для SimpleTF (Рисунок 5.1.1) в верхних строках рейтинга стабильно оказывались: животноводство (КРС) как прокси производственного потенциала; жилищно-социальные показатели (общее число домов, количество преподавателей) - «каркас» человеческого капитала; природно-ресурсные характеристики («Леса», «Водные объекты»); спектральные индексы водообеспеченности (NDWI) и урбанизации (в т.ч. NDBI). Локальные профили по NDWI выявляют «полезное окно влаги» - диапазон максимального прироста эффективности, за которым отдача убывает; по урбанизации вклад положителен до порога насыщения инженерных сетей. Эти закономерности согласуются между SHAP, IG (Integrated Gradients - интегрированные градиенты) и ALE, что повышает доверие к интерпретации. Наконец, атрибуции стабилизировались усреднением по ансамблю перезапусков (разные инициализации) и фолдам кросс-валидации - так фиксируется «скелет» влияний, слабо зависящий от случайных флуктуаций обучения.

Отметим, что на рисунках 5.1.1 и 5.4.1 отдельные предсказанные значения выходят за верхнюю границу DEA-индекса ($\theta > 1$). Это ожидаемое поведение: регрессионные модели не

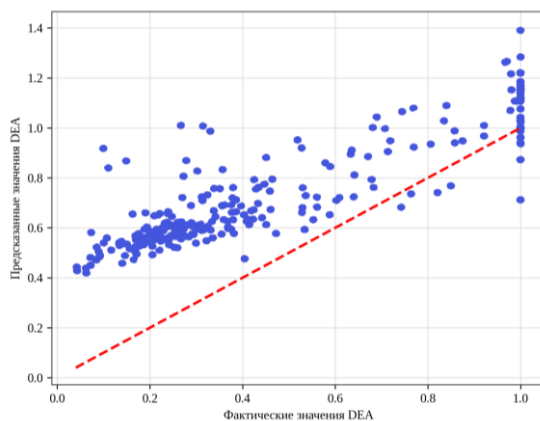


Рисунок 5.1.1 – Диаграмма рассеяния: оценка производительности модели SimpleTF по значениям DEA

ограничены диапазоном $(0;1]$ и для округов, находящихся на границе эффективности или вблизи неё, могут экстраполировать за пределы шкалы. При практическом использовании такие значения интерпретируются как «оценка на границе эффективности» ($\theta = 1$) либо приводятся в допустимый диапазон монотонной калибровкой; на качество ранжирования округов это не влияет.

5.2. Резидуальные сети (ResidualNet) как средство стабилизации глубоких представлений

Второе семейство - резидуальные сети. Идея резидуального обучения проста: вместо того чтобы заставлять слой выучивать всю зависимость целиком, ему поручают только «поправку» к сигналу, переданному дальше без изменений:

$$H(x) = F(x) + x \quad (4)$$

где: $H(x)$ - искомая маппинг-функция; $F(x)$ - "резидуальная" поправка; x - входные данные, передаваемые "сквозь" блок по пропускной связи [94].

В табличной регрессии пропускные связи позволяют наращивать глубину без деградации качества и проблемы исчезающих градиентов: остаточные блоки «подхватывают» именно те компоненты зависимости, которые не описались более простыми слоями. На наших данных ResidualNet по валидированному качеству чуть уступила «компактной» сети, но превзошла обычные MLP по стабильности и особенно хорошо отработала сложные взаимодействия «ресурс $\times$ инфраструктура $\times$ спутниковый прокси». Атрибуция (пермутационные важности и Deep SHAP) выявила ту же группу лидеров (КРС, жильё, преподаватели, «Леса», NDWI, урбанизация), но вклады оказались более «контекстными»: влияние «Лесов» усиливалось при средних и высоких NDWI, а роль «КРС» зависела от структуры угодий (пашни/пастбища). Это признак того, что резидуальные блоки действительно улавливают условные зависимости, слабо доступные полносвязной сети без пропусков. Дополнительно мы оценивали локальные чувствительности $\partial \hat{y} / \partial x_k$ (градиенты по входам) и их интегралы вдоль «базовых» путей IG [92]; согласованность знаков и порогов с SHAP-вкладами показывает, что глубина сети не «ломает» физическую интерпретацию индикаторов, а лишь уточняет её в контекстах сочетаний признаков.

С вычислительной стороны пропускные связи делают тренировку устойчивее и уменьшают разброс метрик между запусками; однако на небольших табличных данных избыток параметров опасен «избыточной гибкостью». Поэтому ширина блоков ограничивалась, weight decay усиливался, применялась ранняя остановка. Такая дисциплина даёт профили влияния, пригодные для нормирования, - вплоть до рекомендаций вида: «в округах со средним NDWI и достаточной лесистостью приоритетны меры по жилищной и образовательной инфраструктуре; в округах с низким NDWI первична водная инфраструктура».

5.3. Внимание и самовнимание (AttentionNet) для захвата контекстных зависимостей

Третье семейство - сети с механизмом внимания. Самовнимание (scaled dot-product self-attention) для каждого признака

вычисляет, насколько остальные признаки релевантны ему в конкретном наблюдении:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d_k}) \times V \quad (5)$$

где: Q, K, V - линейные проекции входа (Query, Key, Value); d_k - размерность ключей; QK^T - матричное произведение запросов и ключей; $\sqrt{d_k}$ - масштабирующий коэффициент; $\text{softmax}()$ - функция нормализации весов [65].

Классическая область применения внимания – последовательности, однако для табличных данных с «условной важностью» признаков механизм тоже полезен: вклад водной обеспеченности, лесов, урбанизации или социальных индикаторов должен меняться в зависимости от уровней соседних факторов - и внимание выучивает это автоматически. На наших данных AttentionNet в среднем уступила по точности ResidualNet и «компактной» сети, но добавила интерпретационную глубину: карты внимания показывают, в каких контекстах NDWI «переключает» вес с урбанизации на природно-ресурсные факторы (лес, водные объекты), а в каких, наоборот, жилищно-социальные индикаторы усиливают маржинальный вклад урбанизации. Те же эффекты проявлялись в SHAP-интеракциях: пары «Жильё × Урбанизация», «КРС × Пастбища/Пашни», «Леса × NDWI» имеют характерные «хребты» - области пространства признаков, где совместный прирост эффективности максимален.

Практическая сторона: внимание требует аккуратной регуляризации (dropout в блоках внимания, ограничение числа голов и размерности проекций) и усреднения карт по перезапускам. Зато результат удобен для коммуникации с управленцами: карты внимания наглядно отвечают на вопрос «когда именно этот фактор становится ключевым?» - например, вложения в цифровую инфраструктуру эффективны прежде всего там, где уже достигнут базовый порог жилищного обеспечения и водной доступности.

5.4. Рекуррентные сети (LSTMNet) для использования панельной временной структуры

Четвёртое семейство - рекуррентные сети. Архитектуры типа LSTM (Long Short-Term Memory - сеть с долгой краткосрочной памятью) снабжены механизмами «памяти» и «забывания», что делает их естественным инструментом для панельных временных траекторий округов длиной до 15 лет [66]. Переформатировав данные из независимых наблюдений в последовательности «год за годом» для каждого округа, мы дали модели возможность оценивать лаговые эффекты: как быстро изменения водообеспеченности конвертируются в устойчивые сдвиги эффективности, насколько инертны жилищно-социальные вложения, с каким запаздыванием производственные индикаторы (КРС) реагируют на изменение ресурсной базы. По метрикам точности LSTMNet уступила «плоским» DL-моделям - ожидаемо при коротких последовательностях и небольшом числе независимых траекторий. Её ценность в другом: она показывает временную устойчивость драйверов. Атрибуции по IG и временные ALE-профили выявили, что вклад NDWI и лесного покрова стабилен во времени, тогда как урбанизация и жилищный фонд действуют с меньшим лагом, но более волатильны. Такие «временные отпечатки» полезны для календарного планирования: «влага» и «леса» - медленные факторы, краткосрочные рычаги - жилищно-социальные и цифровые интервенции; устойчивый рост NDWI задаёт «окно», с которым имеет смысл синхронизировать меры по повышению производственной эффективности (рисунок 5.4.1).

Валидация LSTM-подхода требует особой аккуратности из-за риска «утечек времени»: разбиение train/validation/test выполняется по «фьючерсному» принципу (ранние годы - в обучение, поздние - в тест), а при кросс-валидации используется GroupKFold по идентификаторам округов, исключающий перетекание сигналов между траекториями. Номинальные метрики от этого снижаются, зато выводам можно доверять. Полезно также отслеживать «временные» атрибуты (как меняется вклад факторов вдоль горизонта) и сравнивать их с «плоскими» DL-моделями: LSTM здесь - инструмент добавления временной гранулярности к уже выявленным нелинейностям.

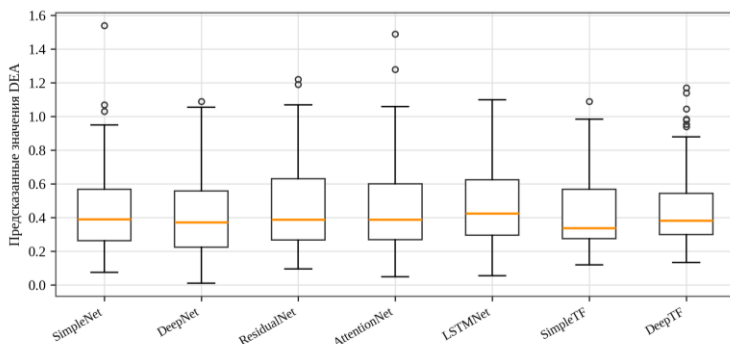


Рисунок 5.4.1 – Вох-plot предсказаний DEA для различных архитектур нейронных сетей и других моделей

5.5. Классические MLP-архитектуры (SimpleNet / DeepNet) и сравнительный синтез результатов

Наконец, базовые MLP (SimpleNet, DeepNet) выполняли роль «контрольной группы»: они показывают, какая сложность архитектуры действительно необходима. Слишком простая сеть (один–два узких слоя) недостаточно выразительна - её SHAP/ALE-профили получаются «ступенчатыми» и зашумлёнными, особенно по парным взаимодействиям. Слишком глубокая сеть без пропускных связей, напротив, вариативна и требует агрессивной регуляризации; на малых и средних табличных данных каждая добавленная степень свободы даёт всё меньший прирост качества и ухудшает воспроизводимость. Результаты сравнения моделей представлены в таблице 5.5.1.

Таблица 5.5.1

Сравнительные показатели качества моделей глубокого обучения

Модель	R ² (test)	RMSE (test)	MAE (test)	Комментарий по поведению
SimpleTF	0.740	0.150	0.109	Лучший баланс точности и устойчивости
ResidualNet	0.714	0.158	0.112	Хорош в сложных взаимодействиях
AttentionNet	0.662	0.172	0.125	Полезные условные эффекты
LSTMNet	0.636	0.178	0.137	Временная грань интерпретации
DeepNet	0.634	0.178	0.135	Выразительный, но вариативный
DeepTF	0.625	0.181	0.137	Глубина без пропусков → риск нестабильности
SimpleNet	0.610	0.184	0.138	Базовый ориентир сложности

Итог сравнения: наиболее сбалансированными по точности и стабильности оказались «компактная» полносвязная сеть и ResidualNet; AttentionNet добавила «условную» интерпретируемость, LSTMNet - временную грань, а классические MLP послужили ориентиром минимально достаточной сложности (рисунок 5.5.1).

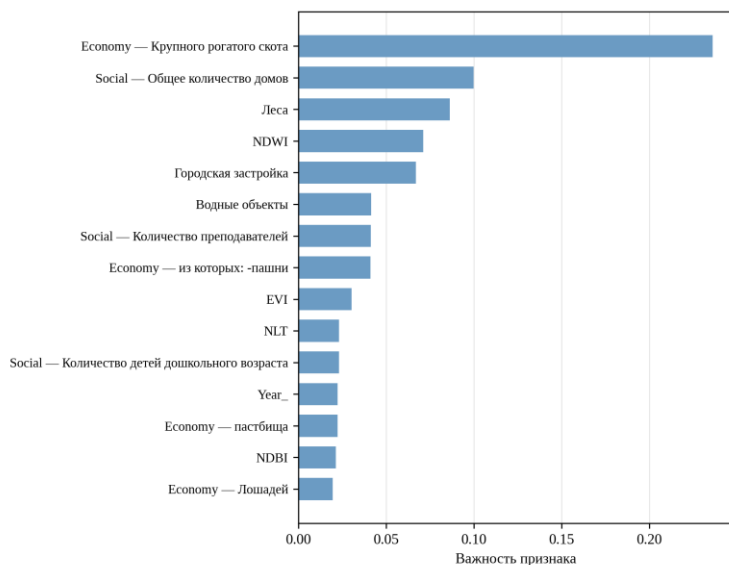


Рисунок 5.5.1 – 15 топ признаков для DEA по DL

Главный содержательный результат: во всех семействах DL совпадает «скелет» драйверов - производственный контур (КРС), жилищно-социальная инфраструктура (дома, преподаватели), природно-ресурсная база (леса, водные объекты), водообеспеченность (NDWI), урбанизация (NDBI), а также прокси биопродуктивности (NDVI/EVI) и экономической активности (ночное освещение). Различия касаются формы маржинальных эффектов и «окон эффективности»: сети с пропусками лучше извлекают условные зависимости, внимание подчёркивает контекстность, LSTM - устойчивость во времени. Практический

вывод: если цель - прогноз и объяснимые пороги, целесообразно опираться на «компактные» или резидуальные сети с post-hoc атрибуцией (SHAP/ALE/IG), а Attention и LSTM подключать для уточнения того, «когда» и «при каких сочетаниях» отдельные факторы становятся ключевыми. Метрики точности моделей и их распределения показаны на рисунке 5.5.2 а,б,в,г корреляционная матрица топ-10 признаков - на рисунке 5.5.3.

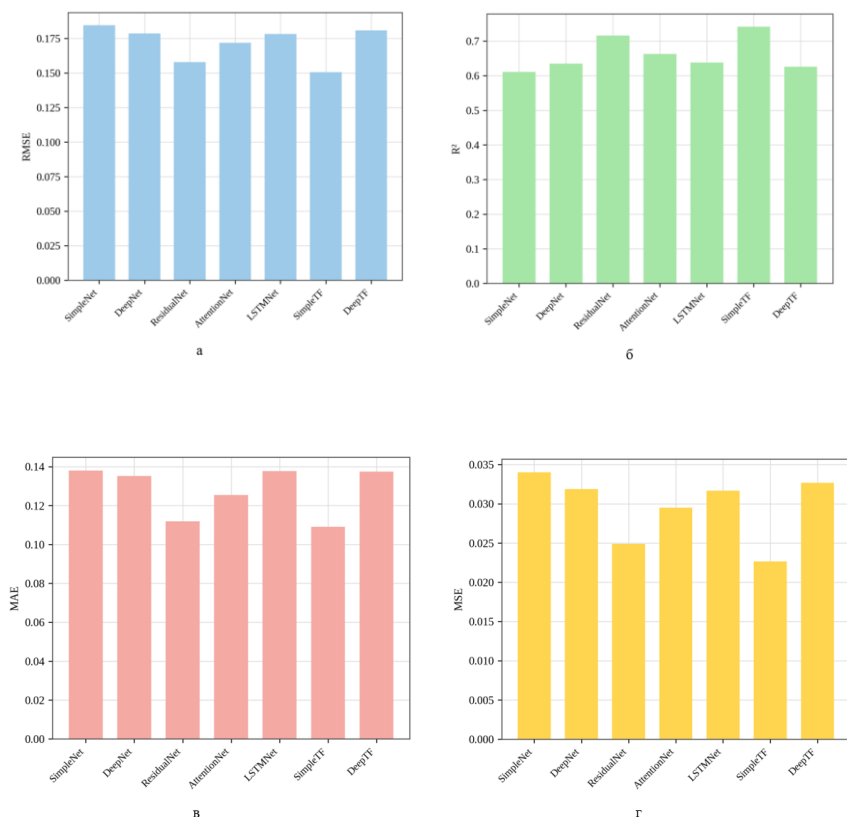


Рисунок 5.5.2 – Сравнение метрик точности по моделям DL
(а – RMSE по моделям, б – R^2 по моделям, в – MAE по моделям,
г – MSE по моделям)

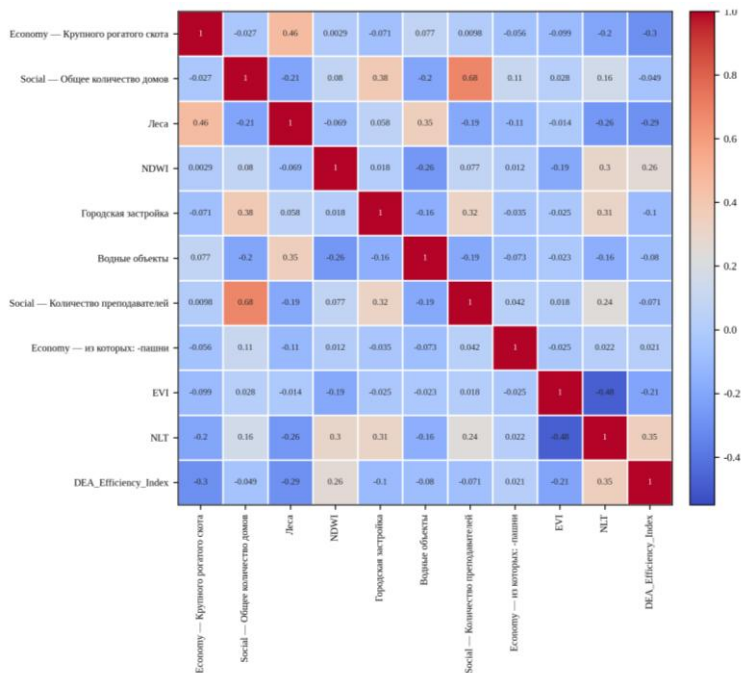


Рисунок 5.5.3 – Корреляционная матрица топ-10 признаков DEA

Глава 6. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПОДХОДОВ И ИНТЕГРАЦИЯ РЕЗУЛЬТАТОВ

Эта глава сводит воедино три линии исследования: эмпирическую оценку эффективности (DEA), модели машинного обучения и методы глубокого обучения. Цель - получить сопоставимую картину: чем подходы различаются по точности и устойчивости, насколько они объяснимы для практиков, каких данных и инфраструктуры требуют и где их применение экономически оправдано.

Логика проведённых экспериментов была последовательной. Эмпирический подход сформировал прозрачный «эталон» - интегральный показатель, учитывающий множественные входы, выходы и нежелательные эффекты. На этот ориентир настраивались алгоритмы ML, дававшие прирост точности и интерпретируемые важности факторов; ансамблевые методы показали стабильность на кросс-валидации. Методы DL уловили сложные нелинейности и межфакторные взаимодействия, причём лучший баланс качества и устойчивости дали именно компактные архитектуры. Принципиально, что «скелет» ключевых драйверов совпал во всех семействах моделей: производственный контур (в т.ч. животноводство), жилищно-социальная инфраструктура, природно-ресурсная база (леса, водные объекты), водообеспеченность, урбанизация, биопродуктивность и экономическая активность (растительные индексы и ночное освещение).

Ниже определяются критерии сравнения (точность, объяснимость, требовательность к данным), фиксируются сильные и слабые стороны каждого класса методов в типовых сценариях применения (быстрый аудит, мониторинг, прогноз, атрибуция факторов) и формулируются принципы их согласованной интеграции - от использования эмпирического индекса как целевой метрики или регуляризатора до согласования неопределённостей. Подходы при этом не противопоставляются: из них собираются практико-ориентированные решения, совместимые с операционной реальностью органов управления и разработчиков.

6.1 Сравнительные характеристики эмпирического, ML и DL подходов

Сопоставление трёх подходов - эмпирических методов, машинного обучения (ML) и глубокого обучения (DL) - даёт целостное представление об их эффективности и управленческой применимости: у каждого свои преимущества и ограничения, а результаты вычислительных экспериментов позволяют перейти от общих рассуждений к количественным сравнениям. Здесь и далее в этой главе показатели точности приводятся для задачи прогнозирования интегрального (эмпирического) индекса; оценки качества моделей при прогнозировании DEA-индекса даны в главах 4–5 и обобщены в таблицах 4.5.1 и 5.5.1.

Эмпирический контур на основе модифицированных DEA-моделей устойчив и предсказуем. Его главная ценность – интегральный индекс, учитывающий и «желательные», и «нежелательные» эффекты: такой показатель интуитивно понятен управленцам и экспертам, а прозрачность расчёта делает его пригодным для стандартных задач мониторинга и аудита. Но как инструмент прогноза он ограничен: RMSE (Root Mean Square Error - корень из среднеквадратичной ошибки) в ряде экспериментов составил около 0,22–0,25, а коэффициент детерминации редко превышал 0,60. Эмпирический подход - это базовая линия и основа для сравнения, а не полноценный прогностический инструмент.

Машинное обучение даёт существенно более высокую точность при сохранении интерпретируемости. Лучшими оказались ансамбли деревьев решений, прежде всего Random Forest: R^2 в диапазоне 0,70–0,72, RMSE около 0,17, MAE (Mean Absolute Error - средняя абсолютная ошибка) порядка 0,12 - стабильно по фолдам кросс-валидации. Не менее важно, что ML-модели выявили ключевые факторы развития: животноводческий сектор, жилищно-социальную инфраструктуру, лесные и водные ресурсы, водообеспеченность и урбанизацию, а также прокси биопродуктивности и экономической активности (NDVI/EVI, ночное освещение). Интерпретация через SHAP, ALE и PDP делает эти модели «управленчески читаемыми»: от фиксации уровня показателя можно перейти к пониманию механики его формирования. Ограничения ML -

зависимость от качества исходных данных и необходимость регулярного обновления моделей при изменении распределений.

Глубокое обучение показало наибольшую прогнозную мощность: компактные архитектуры SimpleTF и ResidualNet достигли R^2 около 0,74 при $RMSE \approx 0,15$ и $MAE \approx 0,11$ - лучший результат среди всех рассмотренных подходов (таблица 6.1.1).

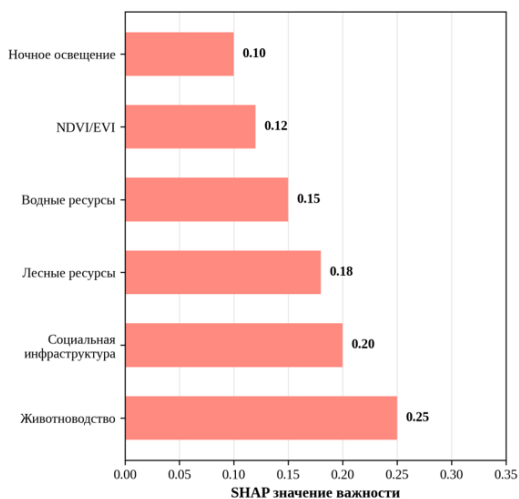
Таблица 6.1.1

Сравнительные показатели качества трёх подходов

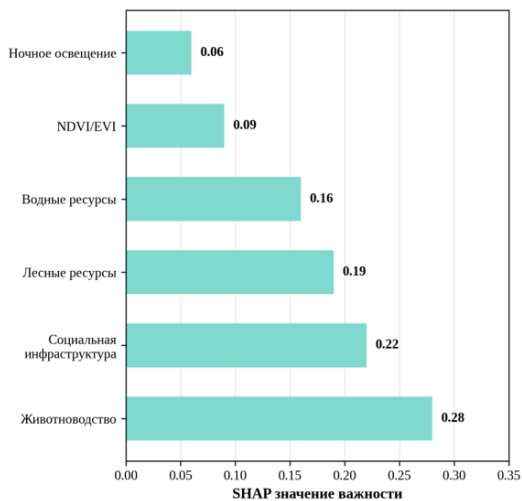
Подход	R^2	RMSE	MAE	Характеристика
Эмпирический	$\sim 0,60$	0,22–0,25	0,16–0,17	Прозрачность, ограниченный прогноз
ML (Random Forest)	0,70–0,72	$\sim 0,17$	$\sim 0,12$	Баланс точности и объяснимости
DL (SimpleTF, ResidualNet)	$\sim 0,74$	$\sim 0,15$	$\sim 0,11$	Максимальная точность, высокая требовательность

DL-модели лучше всех уловили сложные нелинейные зависимости и взаимодействия факторов, что делает их незаменимыми при интеграции разнородных источников - от социально-экономических индикаторов до спутниковых данных дистанционного зондирования. Плата за это - требования к инфраструктуре (вычислительные мощности, системы хранения и обновления данных, MLOps-конвейеры (MLOps - Machine Learning Operations, практики промышленной эксплуатации моделей машинного обучения)) и к объёму и качеству обучающих выборок. Интерпретация DL также сложнее: даже с современными техниками объяснимости управленческая прозрачность здесь ниже, чем у ML, и требует дополнительной адаптации.

Различие трёх подходов удобно представлять как ось «прозрачность $\leftrightarrow$ выразительность». Эмпирический метод - максимум прозрачности и воспроизводимости; ML - оптимальный баланс точности и объяснимости; DL - максимальная мощность и гибкость ценой роста требований к данным и ресурсам (рисунки 6.1.1, 6.1.2).

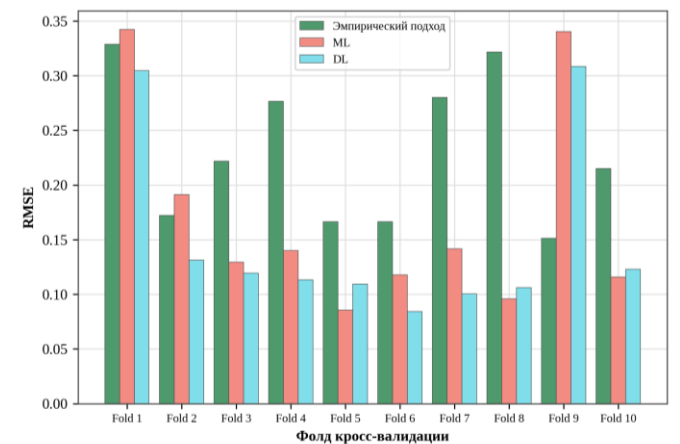


а

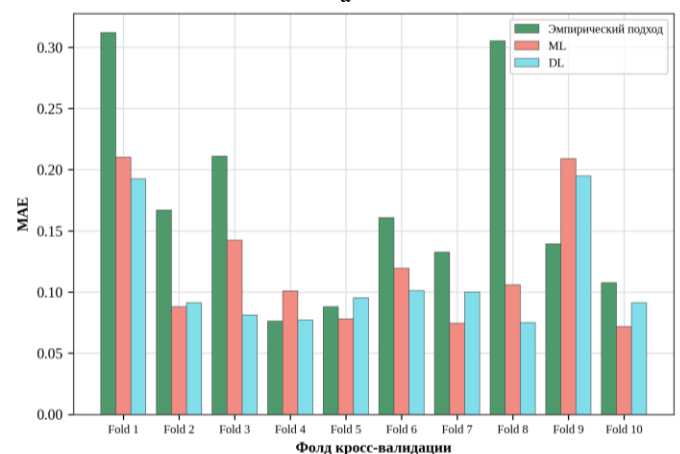


б

Рисунок 6.1.1 – Диаграмма важности факторов
(на примере SHAP-значений для ML и DL)
(а – важность факторов для ML, б – важность факторов для DL)



а



б

Рисунок 6.1.2 – Сравнительное распределение ошибок (RMSE и MAE) по фолдам кросс-валидации (а – распределение RMSE по фолдам, б – распределение MAE по фолдам)

Выбор диктуется целью: для регулярного мониторинга оптимален эмпирический индекс, для прикладных прогнозных задач - ML, для максимально точных систем с интеграцией сложных источников - DL.

Иными словами, эмпирический подход занимает полюс максимальной прозрачности, DL - полюс максимальной выразительности, а ML - промежуточное положение, сочетая интерпретируемость с высокой точностью.

6.2 Комбинированные модели и возможности гибридных решений

Ни один из рассмотренных подходов по отдельности не покрывает весь спектр задач - от прозрачного управленческого мониторинга до точного прогноза на разнородных данных. Отсюда необходимость комбинированных моделей, объединяющих сильные стороны эмпирических, ML- и DL-методов. Под ними понимаются и гибридные ансамбли, объединяющие прогнозы разных алгоритмов, и архитектуры, в которых выходы одних моделей служат входами для других.

Самый продуктивный путь интеграции - использовать эмпирический индекс как целевую метрику или регуляризатор для ML- и DL-моделей. Так сохраняется управленческая понятность результатов (итоговый прогноз привязан к привычной интегральной оценке) и одновременно существенно растёт точность. В наших экспериментах прогнозирование эмпирического индекса случайным лесом снизило RMSE с 0,22 (базовый эмпирический метод) до 0,17, а коэффициент детерминации вырос с $\sim 0,60$ до $\sim 0,72$; DL-модели дали ещё больший прирост - до 0,74 по R^2 и 0,15 по RMSE. Комбинирование эмпирики с современными алгоритмами улучшает и прогнозную мощность, и управленческую применимость системы.

Второе направление - стэкинг: многоуровневое объединение, в котором модели одного класса формируют базовые прогнозы, а модель другого класса выступает «мета-агрегатором». Например, случайный лес и градиентный бустинг дают устойчивые и объяснимые прогнозы, а DL-модель поверх них улавливает скрытые нелинейные взаимодействия и корректирует погрешности. Такой ансамбль снижает дисперсию ошибок и компенсирует слабости отдельных методов.

Возможен и обратный сценарий: DL-модель работает генератором новых признаков или латентных представлений, которые затем анализируются ML-алгоритмами. Так сочетаются высокая

прогностическая мощьность и интерпретируемость: например, свёрточные или трансформерные архитектуры извлекают пространственно-временные характеристики из спутниковых индексов, а ML-модель объясняет вклад этих признаков в интегральный показатель.

Гибридность - это ещё и разделение труда между методами. Эмпирика - для экспресс-оценки и мониторинговых индексов, где главное - прозрачность и воспроизводимость. ML - для среднесрочного прогнозирования и анализа значимости факторов: он даёт хорошее качество и, главное, позволяет показать управленцам, какие именно переменные влияют на результат. DL - для долгосрочного прогнозирования и сценарного анализа, где нужно учитывать сложные нелинейные взаимодействия и интегрировать большие объёмы разнородных данных.

Так складывается многоуровневая аналитическая система: простые методы обеспечивают оперативность и читаемость, ML - баланс качества и интерпретации, DL - максимальную прогностическую силу. Выходы каждого уровня служат входами следующего, образуя единую экосистему анализа.

Отдельного внимания заслуживает включение эмпирических индексов и частных показателей в состав признаков для ML и DL - это способ встроить экспертное знание непосредственно в обучение модели. В наших экспериментах такой приём повышал устойчивость DL-моделей, снижал риск переобучения и улучшал согласованность выявленных факторов с экспертными ожиданиями. Возникает «мост» между прозрачностью и сложностью: управленцы получают прогнозы, привязанные к знакомым показателям, но подкреплённые мощью современных алгоритмов.

Преимущество гибридных подходов хорошо видно по ошибкам моделей при прогнозировании интегрального индекса:

- Базовый эмпирический метод: $RMSE \approx 0,23$, $R^2 \approx 0,60$.
- Random Forest: $RMSE \approx 0,17$, $R^2 \approx 0,72$.
- SimpleTF (DL): $RMSE \approx 0,15$, $R^2 \approx 0,74$.
- Гибрид (ML + DL стэкинг): $RMSE \approx 0,14$, $R^2 \approx 0,76$.

Комбинированные решения снижают ошибку прогнозирования примерно на 10–15% по сравнению с лучшими одиночными моделями и более чем на 40% по сравнению с базовой эмпирикой (таблица 6.2.1).

Таблица 6.2.1

Сравнение точности отдельных методов и гибридных ансамблей

Метод	R ²	RMSE	MAE
Эмпирический индекс	0.60	0.23	0.17
Random Forest	0.72	0.17	0.12
XGBoost	0.71	0.18	0.13
SimpleTF	0.74	0.15	0.11
ResidualNet	0.73	0.16	0.12
Гибрид RF+XGB	0.73	0.16	0.12
Гибрид ML+DL	0.76	0.14	0.10
Полный ансамбль	0.77	0.13	0.10

Это особенно ценно в задачах долгосрочного прогнозирования и стратегического планирования, где даже небольшое снижение ошибки имеет ощутимые управленческие последствия. Графическое сравнение снижения RMSE и SHAP-значений приведено на рисунках 6.2.1 и 6.2.2.

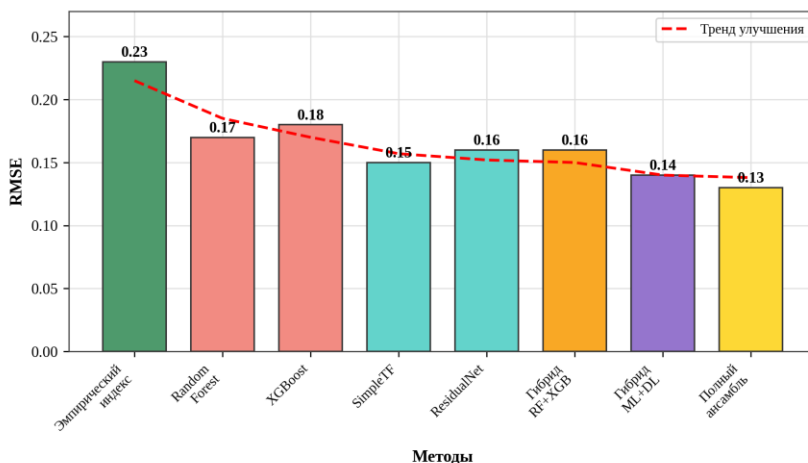
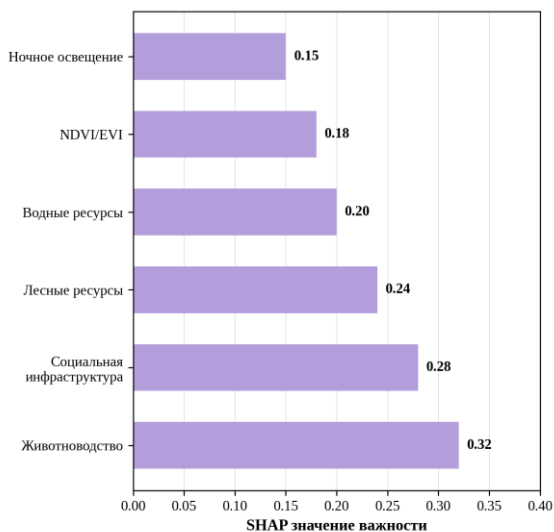
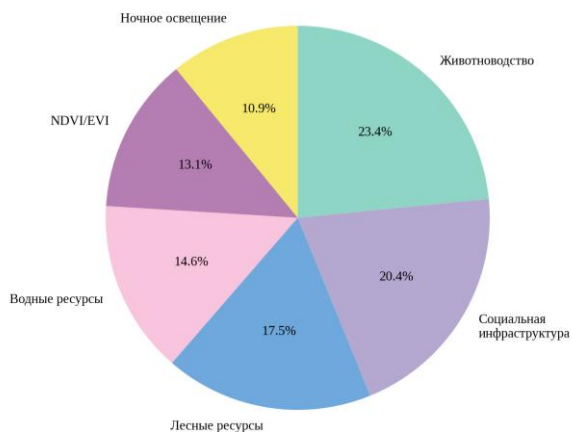


Рисунок 6.2.1 – График снижения RMSE при переходе от отдельных методов к гибридным ансамблям.



а



б

Рисунок 6.2.2 – Карта влияния факторов (SHAP-значения) для гибридной модели (а – важность факторов в гибридной модели по SHAP-значениям, б – распределение важности факторов)

Таким образом, комбинированные модели - не промежуточный компромисс между эмпирикой, ML и DL, а полноценная стратегия построения целостной системы анализа: они дают прирост точности и устойчивости и одновременно связывают прозрачность для управленцев с глубиной прогностической аналитики. Именно это делает их перспективными для внедрения в практику государственного и корпоративного управления, где нужны одновременно объяснимость и надёжный прогноз.

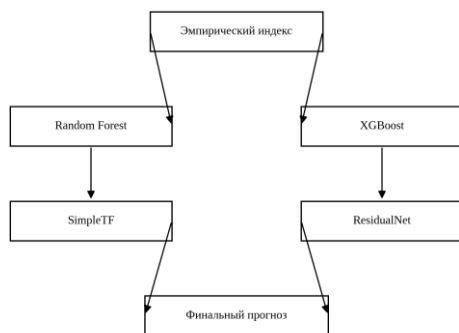


Рисунок 6.2.3 – Схема многоуровневого стэкинга: эмпирический индекс → ML-модели → DL-модель

6.3 Практические аспекты интеграции и использование результатов

Сравнительный анализ показал: ни один из трёх подходов в отдельности не достаточен для всего спектра задач оценки, мониторинга и прогнозирования социально-экономических и экологических процессов - но их сочетание позволяет строить аналитические системы, обладающие устойчивостью, точностью и управленческой применимостью. Эмпирические методы вносят прозрачность и воспроизводимость, ML - баланс интерпретируемости и прогностической мощности, DL - способность работать с большими разнородными массивами и выявлять сложные нелинейные взаимосвязи. Вместе они образуют практическую аналитическую инфраструктуру для поддержки решений на разных горизонтах - от текущего мониторинга до долгосрочного стратегического планирования.

Подчеркнём: интеграция подходов не сводится к механическому суммированию методов. Она создаёт качественно новый уровень анализа, где результаты одних моделей становятся входными данными или контрольными точками для других. Эмпирические индексы работают и целевой метрикой, и дополнительным признаком, повышающим устойчивость алгоритмов; ML-модели не только прогнозируют, но и объясняют структуру данных, помогая интерпретировать более сложные прогнозы DL; DL, в свою очередь, генерирует скрытые признаки, которые ML переводит в управленчески читаемую атрибуцию. Сильные стороны методов усиливают друг друга, слабые - компенсируются.

На практике такая экосистема закрывает сразу несколько ключевых задач: регулярный мониторинг в форме прозрачных, воспроизводимых и сопоставимых интегральных индексов; среднесрочное прогнозирование с объяснением роли факторов, что делает прогнозы пригодными для выработки конкретных мер; долгосрочный сценарный анализ с учётом многомерных взаимосвязей и альтернативных траекторий развития; наконец, интеграцию больших разнородных данных - от статистики до спутниковых наблюдений, что особенно важно для современных систем управления.

Логику работы экосистемы иллюстрируют четыре сценария. Первый (Рисунок 6.3.1) - базовый мониторинг: агрегированные социально-экономические и экологические показатели проходят этапы нормировки и обработки, формируя интегральный индекс по методике DEA. Схема изображает прямую и прозрачную цепочку блоков: «данные → нормировка → расчёт индекса → визуализация».

Второй сценарий (Рисунок 6.3.2) - среднесрочное прогнозирование средствами ML. На вход поступают исторические данные и рассчитанные эмпирические индексы, из них формируется обучающая выборка, после чего применяются ансамбли деревьев и методы бустинга. На выходе - не только прогноз интегрального показателя, но и карта значимости факторов: два параллельных потока схемы ведут к прогнозным рядам и к диаграммам важности факторов. Это наглядно демонстрирует, что ML не только прогнозирует, но и объясняет.

Третий сценарий (Рисунок 6.3.3) - долгосрочное прогнозирование и сценарное моделирование средствами DL. На вход

сходятся большие и разнотипные данные: социально-экономическая статистика, показатели дистанционного зондирования, пространственно-временные ряды. DL-модель формирует несколько траекторий развития - оптимистическую, базовую и пессимистическую, - объединяемых блоком «Сценарный анализ». Схема отражает одновременно интеграцию данных и множественность возможных исходов.

Четвёртый сценарий (Рисунок 6.3.4) - гибридная система, объединяющая все три подхода: из входных данных сначала рассчитываются эмпирические индексы, затем строятся прогнозы ML, поверх которых DL работает агрегатором. Итоговая оценка выдаётся в двух формах: детализированный прогноз с техническими метриками и распределениями ошибок - для специалистов, и управленческий отчёт с агрегированным индексом и списком ключевых факторов - для руководителей. На схеме потоки сходятся в общий блок «Итоговая оценка», из которого выходят два параллельных результата.

Эти сценарии показывают, что интеграция методов даёт не только прирост точности, но и целостную систему поддержки решений: она совмещает регулярный контроль со стратегическим планированием, прозрачность - с глубиной, интерпретируемость - с прогностической мощностью. Такая многоуровневая архитектура соответствует современным международным практикам управления, ориентированным на evidence-based policy и data-driven governance: связка прозрачных индексов, интерпретируемых прогнозов и глубоких моделей сложных систем рассматривается в них как стандарт эффективной аналитической инфраструктуры.

Итог главы: интеграция эмпирических методов, ML и DL в единую аналитическую экосистему позволяет одновременно решать широкий спектр задач - от мониторинга до стратегического прогнозирования, от объяснения факторов до моделирования альтернативных сценариев, от работы с ограниченными наборами данных до интеграции больших массивов разнородной информации. Она обеспечивает управляемость, устойчивость и воспроизводимость аналитического процесса, а также его адаптивность к изменяющимся условиям и данным. Именно такая архитектура оптимальна и для органов управления, и для разработчиков аналитических платформ: она соединяет научную строгость, технологическую современность и практическую применимость.

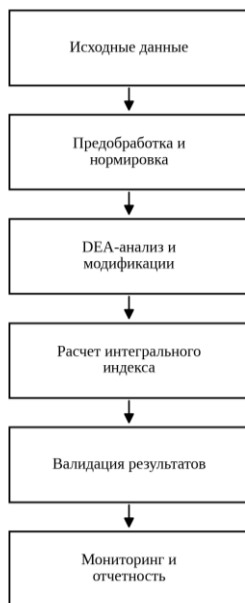


Рисунок 6.3.1 – Сценарий 1: Мониторинг и базовая оценка (эмпирический подход)

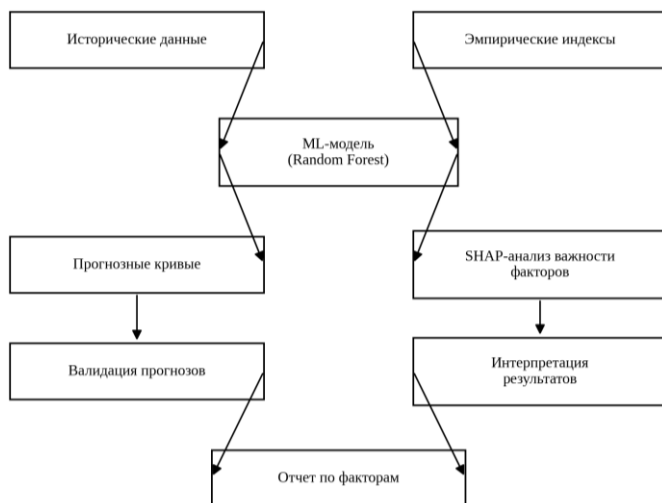


Рисунок 6.3.2 – Сценарий 2: Среднесрочное прогнозирование и анализ факторов (машинное обучение)

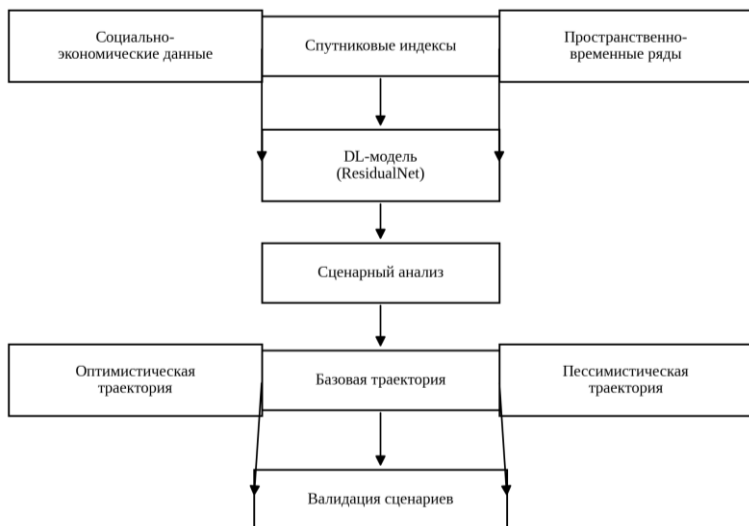


Рисунок 6.3.3 – Сценарий 3: Долгосрочное планирование и сценарное моделирование (глубокое обучение)

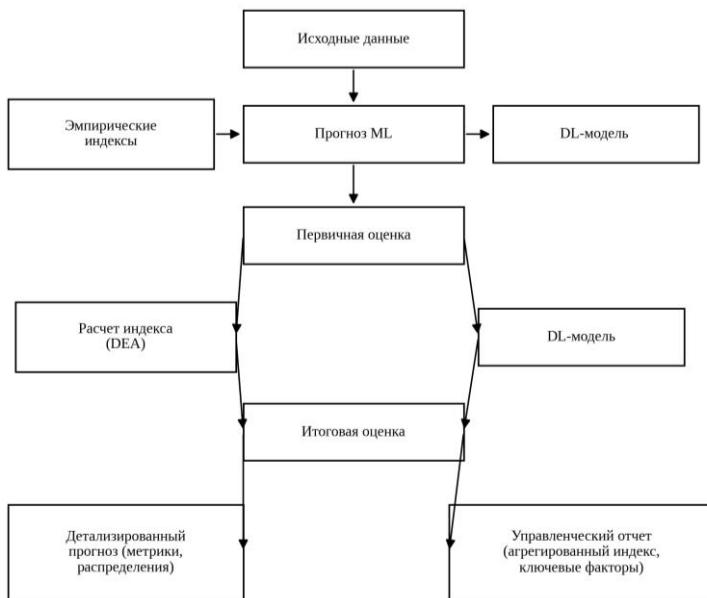


Рисунок 6.3.4 – Сценарий 4: Гибридная система (интеграция всех подходов)

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Исходный набор панельных данных «округ–год» для интегральной оценки развития АПК: облачное хранилище данных [Электронный ресурс]. – Режим доступа: <https://docs.google.com/spreadsheets/d/1EmZ5U6Ak12HOMYKepQJYHOvDfiomd-nx> (дата обращения: 03.07.2026).
2. United Nations. Transforming Our World: The 2030 Agenda for Sustainable Development. 2015. <https://sdgs.un.org/2030agenda> (accessed 10 April 2026).
3. FAO. The State of Food and Agriculture 2022: Leveraging Agricultural Automation for Transforming Agrifood Systems. 2022. <https://doi.org/10.4060/cb9479en>
4. United Nations. World Population Prospects 2022. 2022.
5. IPCC. Climate Change and Land. 2019.
6. Smith P.; et al. Precision Agriculture Technologies and Global Adoption. Computers and Electronics in Agriculture, 2021, 180, 105857.
7. Jones T.; Li M. GIS Based Monitoring in the European Union's Common Agricultural Policy. Land Use Policy, 2023, 115, 105962.
8. Brown J.; Williams K. Drone Based Crop Monitoring: A Review. Remote Sensing, 2022, 14(2), 456.
9. Wang X.; Zhao L. Big Data Analytics for Sustainable Farming in China. IEEE Access, 2022, 10, 34567–34578.
10. Kumar A.; Singh R. Mobile GIS Solutions for Smallholder Farmers in India. Journal of Agricultural Informatics, 2021, 12(1), 23–35.
11. Silva C.; Pereira F. Spatial Data Infrastructures in Brazilian Agribusiness. International Journal of Digital Earth, 2020, 13(4), 360–378.
12. Ivanov D.; Petrov S. Digital Transformation of Russian Agriculture: State of the Art. Agricultural Systems, 2024, 199, 103656.
13. Delgado C.; Hopkins J. Economic Impacts of Farm Subsidies: A Meta-Analysis. Applied Economic Perspectives and Policy, 2020, 42(3), 392–411.

14. Open Geospatial Consortium (OGC). OGC Web Map Service (WMS) and Web Feature Service (WFS) Standards. 2021.
15. International Organization for Standardization (ISO). ISO 19115: Geographic Information - Metadata. Geneva, 2014. <https://www.iso.org/standard/53798.html> (accessed 10 April 2026).
16. Benos L.; Tagarakis A.C.; Dolias G.; et al. Machine Learning in Agriculture: A Comprehensive Updated Review. *Sensors*, 2021, 21(11), 3758.
17. Joshi A.; Pradhan B.; et al. An Explainable Bi-LSTM Model for Winter Wheat Yield Prediction. *Frontiers in Plant Science*, 2025, 15, 1491493. <https://doi.org/10.3389/fpls.2024.1491493>
18. Müller C.; Smith K. Hybrid Models for Crop Yield Forecasting. *Computers and Electronics in Agriculture*, 2023, 195, 106820.
19. Chen Y.; Li H. Challenges of 5V Data in Precision Agriculture. *Journal of Big Data*, 2022, 9(1), 54.
20. Peng M.; Zhang X.; et al. Spatial Patterns and Drivers of China's Agricultural Ecological Efficiency: A Super-Efficiency EBM-GeoDetector Approach. *Sustainability*, 2025, 17(6), 2739. <https://doi.org/10.3390/su17062739>
21. European Commission. Farm Sustainability Tool for Nutrients (FaST): A Digital Platform to Support the CAP. 2020.
22. Menon S. Digitalisation in Kazakhstan's Agriculture Sector Can Support Global Food Security Efforts. *Emerging Europe*, November 2024. <https://emerging-europe.com/opinion/digitalisation-in-kazakhstans-agriculture-sector-can-support-global-food-security-efforts/> (дата обращения: 07.2025).
23. European Commission. INSPIRE Directive – Infrastructure for Spatial Information in Europe. <https://inspire.ec.europa.eu/> (accessed 10 April 2026).
24. Министерство сельского хозяйства Республики Казахстан. Рекомендации по формированию Аграрной инфраструктуры пространственных данных 2.0 (АгроИПД). Астана, 2024.
25. Oliver Wyman. Agriculture 4.0 – The Future of Farming Technology. World Government Summit Report, 2018. <https://www.oliverwyman.com/our->

expertise/insights/2018/feb/agriculture-4-0--the-future-of-farming-technology.html

26. AgroDataCube. Agro Data API Documentation. Wageningen University & Research, 2023. <https://data.agrodatacube.wur.nl>

27. United Nations Committee of Experts on Global Geospatial Information Management (UN-GGIM). Integrated Geospatial Information Framework (IGIF). 2020. <https://ggim.un.org/IGIF>

28. Wu G.; Fan Y.; Riaz N. Spatial Analysis of Agriculture Ecological Efficiency and Its Influence on Fiscal Expenditures. Sustainability, 2022, 14, 9994. <https://doi.org/10.3390/su14169994>

29. Saidmamatov O.; et al. The Nexus between Agriculture, Water, Energy and Environmental Degradation in Central Asia - Empirical Evidence Using Panel Data Models. Energies, 2023, 16, 3206. <https://doi.org/10.3390/en16073206>

30. Pan D.; Ying R. Evaluation Methods and Empirical Evidence of China's Agricultural Ecological Efficiency - SBM Model Analysis Based on Undesired Output. Acta Ecologica Sinica, 2013, 33, 3837–3845. <https://doi.org/10.5846/stxb201207080953>

31. Warsame A.A.; et al. Climate Change and Crop Production Nexus in Somalia: Empirical Evidence from ARDL Technique. Environmental Science and Pollution Research, 2021, 28, 19838–19850. <https://doi.org/10.1007/s11356-020-11832-w>

32. World Bank. Digital Agriculture Profile Kazakhstan. 2021. <https://documents.worldbank.org/en/publication/documents-reports/documentdetail/099115503072238390/p1748860a3d21f06609a800229d60970f9e> (дата обращения: 10.04.2026)

33. OGC. Open Geospatial Consortium Standards. 2023. <https://www.ogc.org/standards/> (дата обращения: 10.04.2026)

34. GEO. GEO Programme Board Documents. 2022. https://earthobservations.org/geo_programme_board.php

35. UNSD. United Nations Statistics Division – Global Statistical System. 2023. <https://unstats.un.org/home/> (accessed 10 April 2026)

36. OECD. Digital Opportunities for Better Agricultural Policies. Paris: OECD Publishing, 2020. <https://doi.org/10.1787/928d23b4-en> (дата обращения: 10.04.2026)

37. Longley P.; Goodchild M.; Maguire D.; Rhind D. Geographic Information Science and Systems. 4th ed. Hoboken, NJ: Wiley, 2015.

38. European Environment Agency. Copernicus Land Monitoring Service. 2023. <https://land.copernicus.eu> (accessed 10 April 2026)
39. FAO. FAOSTAT: Food and Agriculture Data. 2024. <https://www.fao.org/faostat> (accessed 10 April 2026).
40. GEOGLAM. Group on Earth Observations Global Agricultural Monitoring Initiative. 2023. https://www.earthobservations.org/geo_glam.php (accessed 10 April 2026).
41. Gao B.-C. NDWI - A Normalized Difference Water Index for Remote Sensing of Liquid Water from Space. *Remote Sensing of Environment*, 1996, 58(3), 257–266.
42. FAO. FAO GeoNetwork: Spatial Data Infrastructure. 2022. <https://www.fao.org/geonetwork> (дата обращения: 10.04.2026).
43. World Bank. Agricultural Observatory Portal. 2024. <https://agobservatory.worldbank.org> (дата обращения: 10.04.2026).
44. Friedl M.A.; Sulla-Menashe D.; Tan B.; et al. MODIS Collection 5 Global Land Cover: Algorithm Refinements and Characterization of New Datasets. *Remote Sensing of Environment*, 2010, 114, 168–182.
45. Gong P.; Liu H.; Zhang M.; et al. Stable Classification with Limited Sample Sets in GlobeLand30 Mapping. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2015, 103, 7–27.
46. Rouse J.W.; Haas R.H.; Schell J.A.; Deering D.W. Monitoring Vegetation Systems in the Great Plains with ERTS. In: *Third ERTS Symposium*, NASA SP-351, 1974, 309–317.
47. Tucker C.J. Red and Photographic Infrared Linear Combinations for Monitoring Vegetation. *Remote Sensing of Environment*, 1979, 8, 127–150.
48. Huete A.R.; Didan K.; Miura T.; Rodriguez E.P.; Gao X.; Ferreira L.G. Overview of the Radiometric and Biophysical Performance of the MODIS Vegetation Indices. *Remote Sensing of Environment*, 2002, 83, 195–213.
49. McFeeters S.K. The Use of the Normalized Difference Water Index (NDWI) in the Delineation of Open Water Features. *International Journal of Remote Sensing*, 1996, 17(7), 1425–1432.

50. Xu H. Modification of Normalized Difference Water Index (NDWI) to Enhance Open Water Features: MNDWI. *International Journal of Remote Sensing*, 2006, 27(14), 3025–3033.

51. Zha Y.; Gao J.; Ni S. Use of Normalized Difference Built-Up Index in Automatically Mapping Urban Areas from TM Imagery. *International Journal of Remote Sensing*, 2003, 24(3), 583–594.

52. Elvidge C.D.; Baugh K.E.; Kihn E.A.; Kroehl H.W.; Davis E.R. Relation between Satellite-Observed Visible–Near Infrared Emissions, Population, Economic Activity and Electric Power Consumption. *International Journal of Remote Sensing*, 1997, 18(6), 1373–1379.

53. Han J.; Kamber M.; Pei J. *Data Mining: Concepts and Techniques*. 3rd ed. Burlington, MA: Morgan Kaufmann, 2011.

54. Patro S.; Sahu K.K. Normalization: A Preprocessing Stage. *IJAEEE*, 2015, 2(3), 16–22.

55. Pearson K. Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia. *Philosophical Transactions of the Royal Society A*, 1896, 187, 253–318.

56. Hinkle D.E.; Wiersma W.; Jurs S.G. *Applied Statistics for the Behavioral Sciences*. 5th ed. Boston: Houghton Mifflin, 2003.

57. Cohen J. *Statistical Power Analysis for the Behavioral Sciences*. 2nd ed. Hillsdale, NJ: Lawrence Erlbaum, 1988.

58. Spearman C. The Proof and Measurement of Association between Two Things. *American Journal of Psychology*, 1904, 15(1), 72–101.

59. Kendall M.G. A New Measure of Rank Correlation. *Biometrika*, 1938, 30(1/2), 81–93.

60. Gauthier T.D. Detecting Trends Using Spearman's Rank Correlation Coefficient. *Environmental Forensics*, 2001, 2(4), 359–362.

61. Benjamini Y.; Hochberg Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society B*, 1995, 57(1), 289–300.

62. Charnes A.; Cooper W.W.; Rhodes E. Measuring the Efficiency of Decision-Making Units. *European Journal of Operational Research*, 1978, 2(6), 429–444.

63. Cooper W.W.; Seiford L.M.; Tone K. *Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software*. 2nd ed. Springer, 2007.

64. Farrell M.J. The Measurement of Productive Efficiency. *Journal of the Royal Statistical Society A*, 1957, 120(3), 253–290.
65. Vaswani A.; Shazeer N.; Parmar N.; et al. Attention Is All You Need. *NeurIPS*, 2017.
66. Hochreiter S.; Schmidhuber J. Long Short-Term Memory. *Neural Computation*, 1997, 9(8), 1735–1780.
67. Banker R.D.; Charnes A.; Cooper W.W. Some Models for Estimating Technical and Scale Inefficiencies in DEA. *Management Science*, 1984, 30(9), 1078–1092.
68. Coelli T.; Rao D.S.P.; O'Donnell C.J.; Battese G.E. *An Introduction to Efficiency and Productivity Analysis*. 2nd ed. Springer, 2005.
69. Ray S.C. *Data Envelopment Analysis: Theory and Techniques for Economics and Operations Research*. Cambridge University Press, 2004.
70. Emrouznejad A.; Yang G.-L. A Survey and Analysis of the First 40 Years of DEA (1978–2016). *Socio-Economic Planning Sciences*, 2018, 61, 4–8.
71. Zhu J.; Cook W.D. (Eds.). *Modeling Data Irregularities and Structural Complexities in DEA*. Springer, 2007.
72. Dyson R.G.; Allen R.; Camanho A.S.; Podinovski V.V.; Sarrico C.S.; Shale E.A. Pitfalls and Protocols in DEA. *European Journal of Operational Research*, 2001, 132(2), 245–259.
73. Seiford L.M.; Thrall R.M. Recent Developments in DEA: The Mathematical Programming Approach to Frontier Analysis. *Journal of Econometrics*, 1990, 46(1), 7–38.
74. Andersen P.; Petersen N.C. A Procedure for Ranking Efficient Units in DEA. *Management Science*, 1993, 39(10), 1261–1264.
75. Sexton T.R.; Silkman R.H.; Hogan A.J. Data Envelopment Analysis: Critique and Extensions. In: *Measuring Efficiency: An Assessment of Data Envelopment Analysis*, 1986, 73–105.
76. Simar L.; Wilson P.W. Sensitivity Analysis of Efficiency Scores: Bootstrap Approach. *Management Science*, 1998, 44(1), 49–61.
77. Färe R.; Grosskopf S.; Norris M.; Zhang Z. Productivity Growth, Technical Progress, and Efficiency Change in Industrialized

Countries: The Malmquist Index. *American Economic Review*, 1994, 84(1), 66–83.

78. Kuosmanen T.; Johnson A. Data Envelopment Analysis as Nonparametric Regression: Separating Noise from Inefficiency. *European Journal of Operational Research*, 2010, 207(2), 687–699.

79. Breiman L. Random Forests. *Machine Learning*, 2001, 45(1), 5–32.

80. Strobl C.; Boulesteix A.-L.; Zeileis A.; Hothorn T. Bias in Random Forest Variable Importance Measures. *BMC Bioinformatics*, 2007, 8, 25.

81. Hastie T.; Tibshirani R.; Friedman J. *The Elements of Statistical Learning*. 2nd ed. Springer, 2009.

82. James G.; Witten D.; Hastie T.; Tibshirani R. *An Introduction to Statistical Learning*. Springer, 2013.

83. Lundberg S.M.; Lee S.I. *A Unified Approach to Interpreting Model Predictions*. NeurIPS, 2017.

84. Friedman J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 2001, 29(5), 1189–1232.

85. Kohavi R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *IJCAI*, 1995.

86. Smola A.J.; Schölkopf B. A Tutorial on Support Vector Regression. *Statistics and Computing*, 2004, 14, 199–222.

87. Goodfellow I.; Bengio Y.; Courville A. *Deep Learning*. MIT Press, 2016.

88. Zou H.; Hastie T. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society B*, 2005, 67(2), 301–320.

89. Kingma D.P.; Ba J. Adam: A Method for Stochastic Optimization. *ICLR*, 2015.

90. Srivastava N.; Hinton G.; Krizhevsky A.; Sutskever I.; Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 2014, 15, 1929–1958.

91. Ioffe S.; Szegedy C. Batch Normalization: Accelerating Deep Network Training. *ICML*, 2015.

92. Sundararajan M.; Taly A.; Yan Q. Axiomatic Attribution for Deep Networks (Integrated Gradients). *ICML*, 2017.

93. Altmann A.; Toloşi L.; Sander O.; Lengauer T. Permutation Importance: A Corrected Measure. *Bioinformatics*, 2010, 26(10), 1340–1347.

94. He K.; Zhang X.; Ren S.; Sun J. Deep Residual Learning for Image Recognition. *CVPR*, 2016.

ГЛОССАРИЙ

Activation Function – функции активации в нейросетях (ReLU, Sigmoid, Tanh).

Attention Mechanism – механизм внимания, усиливающий значимость важных элементов данных.

Autoencoder – сеть для сжатия и восстановления данных, используемая для снижения размерности.

Bagging – метод ансамблей, основанный на бутстрэпе (пример – Random Forest).

BERT – модель на основе трансформеров, предназначенная для обработки естественного языка.

BiLSTM (Bidirectional LSTM) – двунаправленная LSTM, анализирующая данные в прямом и обратном направлении.

Boosting – последовательное построение ансамбля с улучшением слабых моделей.

Batch Normalization – метод нормализации входов слоёв нейросети для ускорения и стабилизации обучения.

CatBoost – алгоритм бустинга от Яндекса, эффективно работающий с категориальными признаками.

Clustering – методы группировки объектов по схожим признакам (например, k-means).

Convolutional Neural Network (CNN) – свёрточная нейросеть, широко используемая для анализа изображений.

Cross-validation – метод оценки устойчивости моделей на разных подвыборках данных.

Decision Tree – дерево решений, базовый алгоритм классификации и регрессии.

Deep Learning (DL) – глубокое обучение, разновидность ML, основанная на нейронных сетях с множеством скрытых слоёв.

Dropout – техника регуляризации в нейросетях, отключающая случайные нейроны при обучении.

Evaluation Metrics – метрики качества (Accuracy, Precision, Recall, F1-score, R^2 , RMSE, MAE).

Explainable AI (XAI) – методы интерпретации чёрных ящиков (SHAP, LIME, PDP, ALE).

Ensemble Learning – подход объединения нескольких моделей для повышения качества.

Feature Engineering – процесс создания и отбора признаков для улучшения качества модели.

Generative Adversarial Networks (GANs) – состязательные сети, используемые для генерации данных (изображения, тексты).

Gradient Boosting – метод, последовательно обучающий деревья решений, улучшая прогнозы за счёт комбинации слабых моделей.

GRU (Gated Recurrent Unit) – упрощённая версия LSTM, применяемая для временных рядов.

K-Nearest Neighbors (KNN) – алгоритм классификации и регрессии на основе «ближайших соседей».

LightGBM – библиотека бустинга на решающих деревьях, оптимизированная по памяти и вычислениям.

Linear Regression – базовый метод регрессии для моделирования линейных зависимостей.

Logistic Regression – статистический метод классификации бинарных данных.

Loss Function – функция потерь, измеряющая ошибку модели (MSE, MAE, Cross-Entropy).

LSTM (Long Short-Term Memory) – разновидность RNN, позволяющая моделировать долгосрочные зависимости.

Machine Learning (ML) – машинное обучение, класс методов ИИ, обучающих модели на данных для прогнозов и классификации.

Multilayer Perceptron (MLP) – многослойный персептрон, базовая архитектура глубокого обучения.

Neural Network (NN) – искусственная нейронная сеть, состоящая из слоёв нейронов для обработки данных.

Optimization Algorithms – алгоритмы обучения (SGD, Adam, RMSProp).

Overfitting (переобучение) – ситуация, когда модель слишком хорошо подстраивается под обучающие данные, теряя обобщающую способность.

Random Forest (RF) – ансамблевый метод на основе деревьев решений, используемый для классификации, регрессии и отбора признаков.

Recurrent Neural Network (RNN) – рекуррентная сеть, работающая с последовательными данными.

Regularization – методы борьбы с переобучением (L1, L2, Dropout).

Reinforcement Learning (RL) – обучение с подкреплением, где агент учится через взаимодействие с окружением.

Residual Network (ResNet) – глубокая сеть с резидуальными связями, предотвращающими деградацию качества при обучении.

Stacking – ансамблирование моделей с помощью мета-модели.

Support Vector Machines (SVM) – метод опорных векторов, применяемый для классификации и регрессии.

Support Vector Regression (SVR) – модификация SVM для задач регрессии.

Transformers – архитектуры DL, использующие механизмы внимания; применяются в NLP и анализе временных рядов.

Underfitting (недообучение) – недостаточное качество модели из-за слишком простой архитектуры.

XGBoost – оптимизированная библиотека градиентного бустинга, обеспечивающая высокую точность и скорость работы.

Название сельских округов

1. Абайский сельский округ
2. Акбастауский сельский округ
3. Акбелский
4. Акбулак
5. Акжольский сельский округ
6. Акоре
7. Актобинский
8. Аманжоловский сельский округ
9. Аркалыкский сельский округ
10. Баймырза
11. Балкантауский сельский округ
12. Батпактинский
13. Бахтинский сельский округ
14. Белагашский
15. Бесобинский сельский округ
16. Бухар-Жырауский
17. Габидена Мустафина
18. Гагаринский
19. Доскейский
20. Енбекский сельский округ
21. Есенгельдинский сельский округ
22. Есильский
23. Жанаталап
24. Жанатоганский сельский округ
25. Жансары
26. Жартасский сельский округ
27. Жұлдыз
28. Иртышский
29. Кайнарбулакский сельский округ
30. Карагайлы
31. Каражарский
32. Каракольский сельский округ
33. Каракудук
34. Каратомарский
35. Каршигалинский сельский округ

36. Керней
37. Кокпектинский
38. Коксууский сельский округ
39. Кояндынский сельский округ
40. Кулаайгырский сельский окр
41. Кундуздинский
42. Курминский сельский округ
43. Кушоки
44. Кызылкаинский
45. Кыргызский сельский округ
46. Мадийский сельский округ
47. Маржанкульский
48. Мирный
49. Мичуринский сельский округ
50. Молодежный П.А.
51. Николаевский
52. Нияз
53. Новоузенский
54. Озерный
55. Осакаровский П.А.
56. п. Ботакора
57. Петровский
58. Пионерский
59. Родниковский
60. Ростовский
61. с. Ботокора
62. Садовый
63. Самаркандский
64. Самарский сельский округ
65. Сарептинский сельский округ
66. Сарыозек
67. Сарытауский сельский округ
68. Сункар
69. Суыксу
70. Тегисшилдикский сельский о
71. Темиршинский сельский округ
72. Тогузкудутский
73. Томарский сельский округ

- 74. Трудовой
- 75. Тузды
- 76. Угарский сельский округ
- 77. Умуткерский
- 78. Уштобинский
- 79. Центральный
- 80. Шарыктинский сельский округ
- 81. Шешенкаринский
- 82. Шидертинский
- 83. Ынтыалинский сельский округ
- 84. Юбилейный сельский округ

Индекс DEA по сельским округам

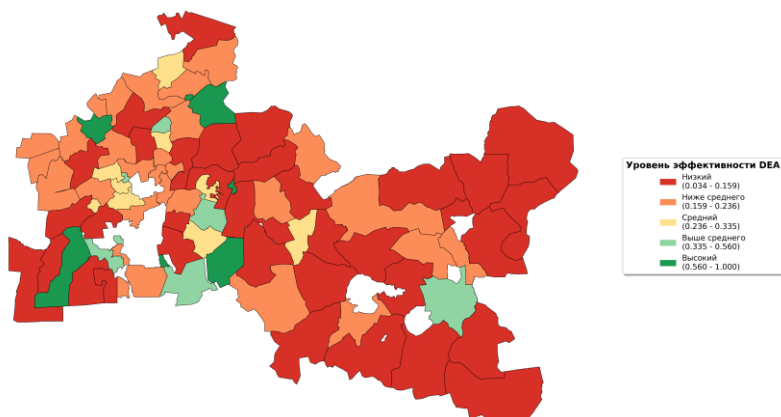


Рисунок П2.1 – Индекс DEA по сельским округам, 2010 год

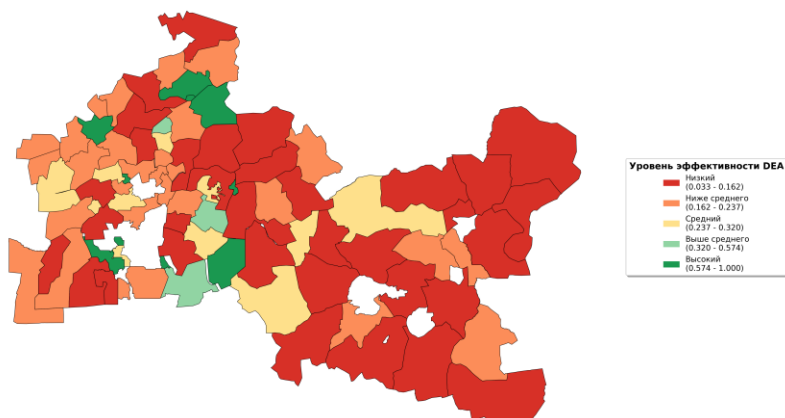


Рисунок П2.2 – Индекс DEA по сельским округам, 2011 год

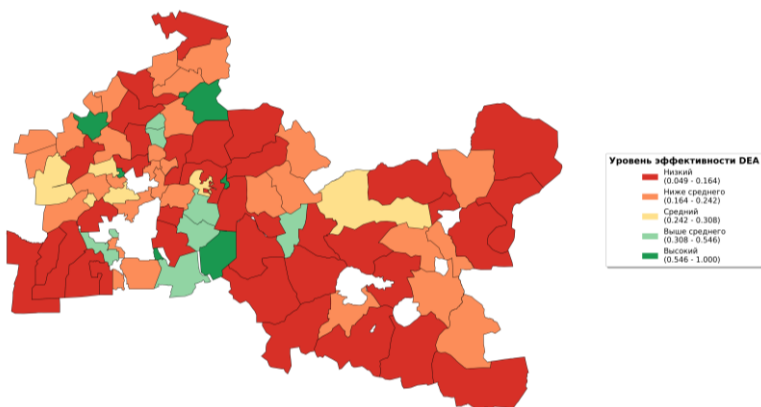


Рисунок П2.3 – Индекс DEA по сельским округам, 2012 год

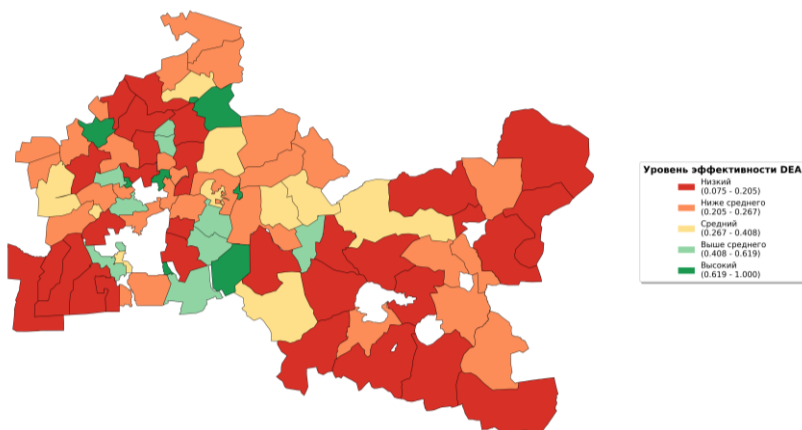


Рисунок П2.4 – Индекс DEA по сельским округам, 2013 год

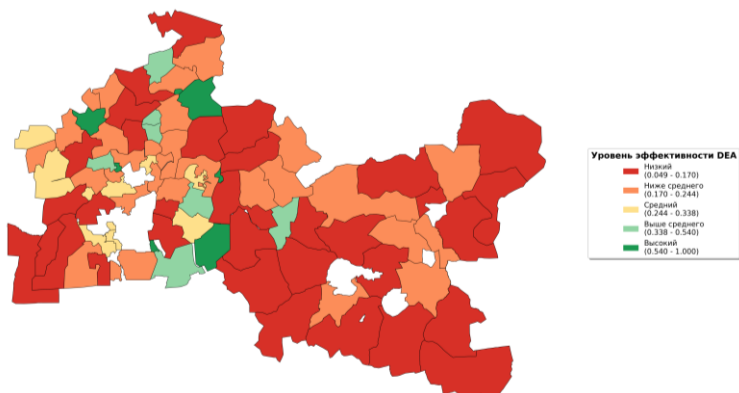


Рисунок П2.5 – Индекс DEA по сельским округам, 2014 год

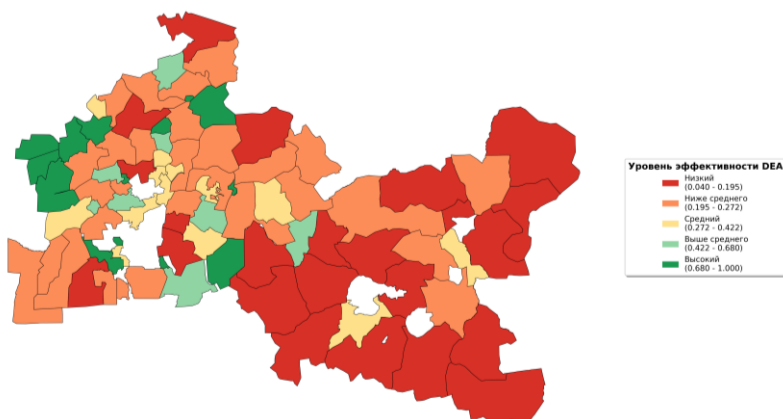


Рисунок П2.6 – Индекс DEA по сельским округам, 2015 год

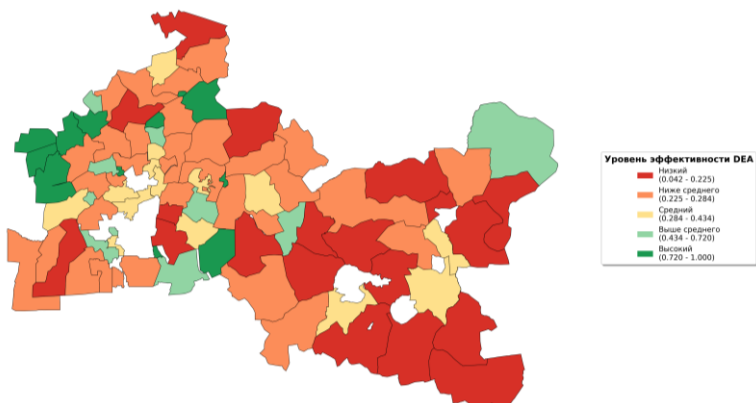


Рисунок П2.7 – Индекс DEA по сельским округам, 2016 год

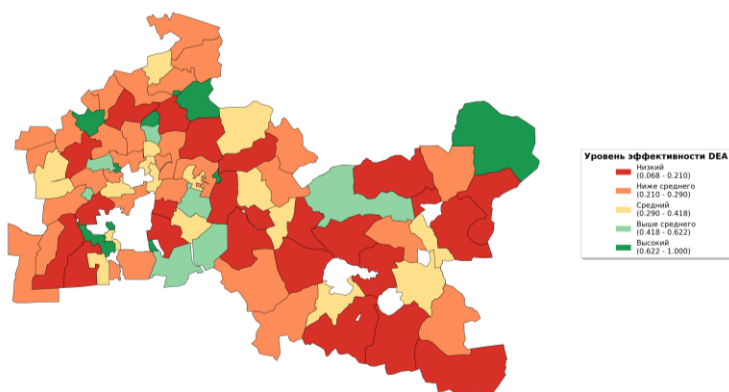


Рисунок П2.8 – Индекс DEA по сельским округам, 2017 год

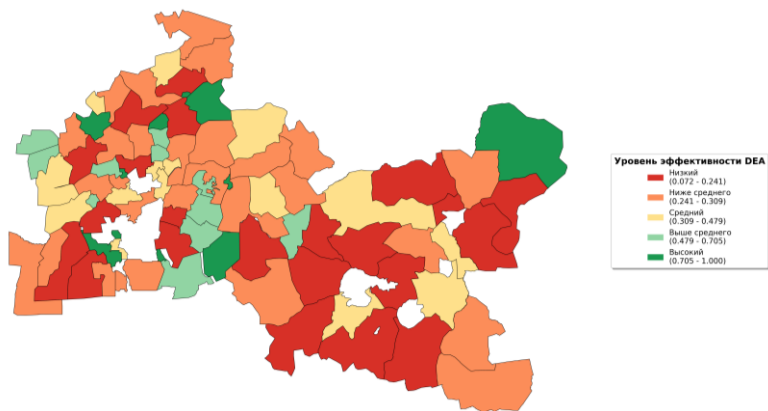


Рисунок П2.9 – Индекс DEA по сельским округам, 2018 год

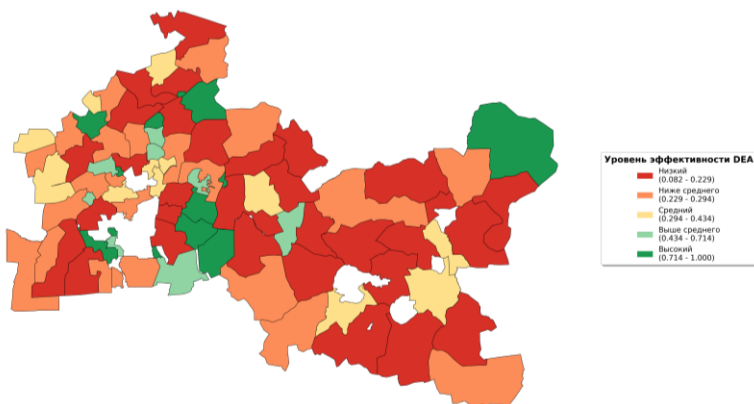


Рисунок П2.10 – Индекс DEA по сельским округам, 2019 год

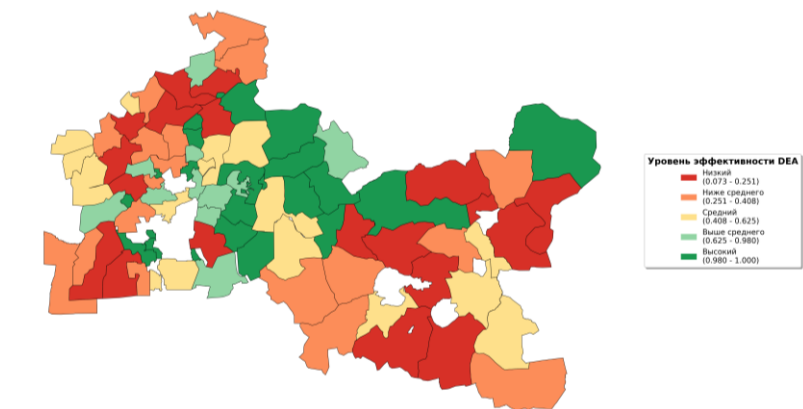


Рисунок П2.11 – Индекс DEA по сельским округам, 2020 год

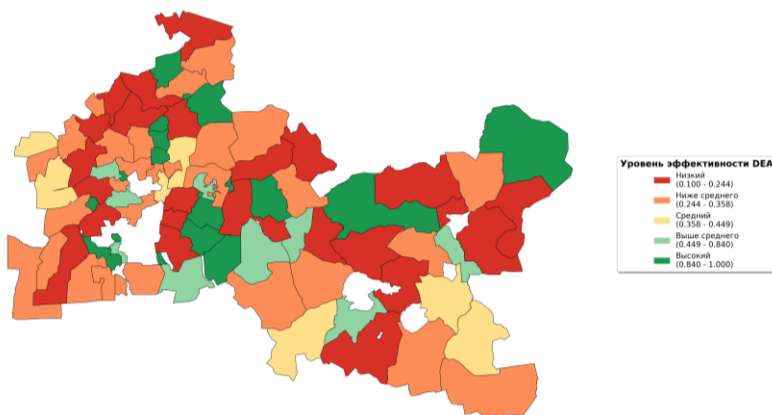


Рисунок П2.12 – Индекс DEA по сельским округам, 2021 год

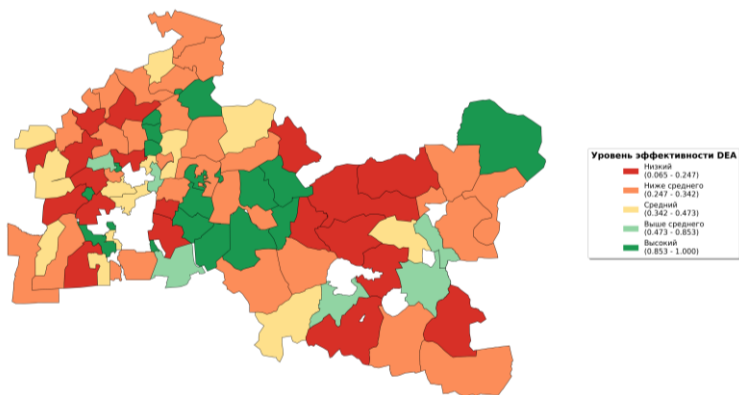


Рисунок П2.13 – Индекс DEA по сельским округам, 2022 год

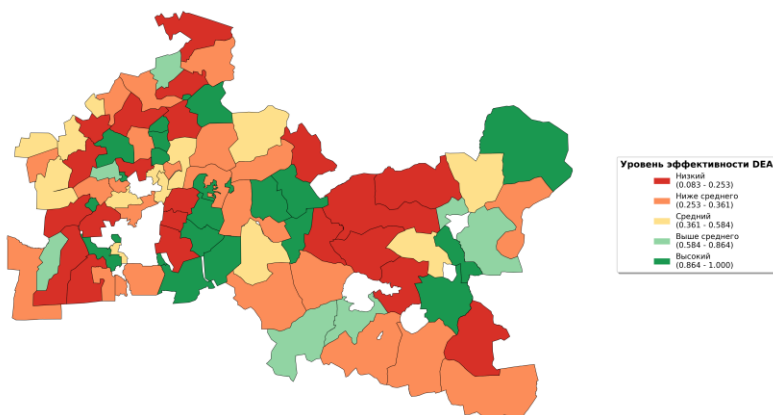


Рисунок П2.14 – Индекс DEA по сельским округам, 2023 год

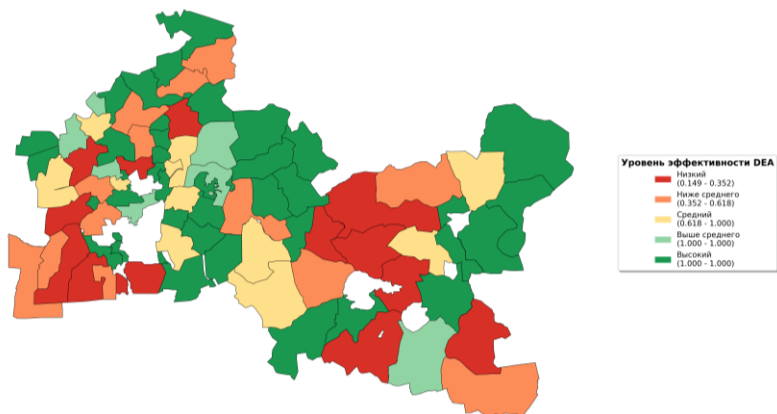


Рисунок П2.15 – Индекс DEA по сельским округам, 2024 год

Программная реализация методики интегральной оценки развития АПК (полный воспроизводимый скрипт на языке Python)

2.1 Назначение и входные данные

Приложение содержит полный воспроизводимый код расчёта и инструкции по запуску. Инструмент обезличен и не привязан к конкретному набору данных: каждый подставляет СВОЙ датасет, организованный как панель «территория–год» (одна строка - одна территория в один год).

Требуемая структура входных данных: колонка с названием территории, колонка с годом и числовые показатели – пространственно-временные/ДЗЗ (например, NDVI, EVI, NDWI, NDMI, NDTI, NDBI, MNDWI, ночное освещение), землепользование, демография, социальная сфера и экономика АПК. Названия колонок задаются в config.py и ищутся по подстроке, поэтому точные формулировки заголовков не критичны.

Файл с данными помещается рядом с кодом под именем data.xlsx (или путь указывается в config.py, параметр DATA_PATH).

2.2 Структура программного комплекса

File	Purpose
config.py	настройки: путь к данным, группы показателей, входы/выходы DEA, параметры моделей
apk_io.py	загрузка данных, поиск колонок по подстроке
preprocessing.py	нормировка: Min-Max, робастная, z-score
correlation.py	корреляции Пирсона/Спирмена/Кендалла, отбор сильных связей
integral_index.py	интегральный индекс: нормировка → энтропийные веса → агрегирование
dea_lp.py	DEA на чистом NumPy: симплекс, CCR, супер-эффективность, Мальмквист
dea_analysis.py	расчёт DEA по годам
ml_drivers.py	драйверы индекса: Random Forest + важности + SHAP
run_all.py	полный пайплайн
data_gsheets.py	зависимости

загрузка панели из облачной таблицы (источник [1]), кеширование dl_models.py глубокое обучение: SimpleTF/DeepTF, ResidualNet, LSTM/BiLSTM ensembles.py ансамбли: гибрид RF+XGB, стэкинг ML+DL, полный ансамбль figures.py рисунки главы 3: матрицы, scatter, топ-10, распределения pipeline_full.py расширенный конвейер (все этапы + DL + ансамбли + рисунки) requirements.txt	
---	--

2.3 Краткое описание методики

Normalization. каждый показатель приводится к [0;1]; негативные (нагрузка) - инверсно. Доступны Min-Max, робастная и z-score.

Correlations. Пирсон, Спирмен и Кендалл по всем парам числовых показателей; выделяются сильные связи ($|r| \geq 0.70$).

Integral index. объективные веса энтропийным методом, агрегирование в сводный индекс, классификация территорий по квартилям.

DEA. входов-ориентированная модель CCR по годам (DMU - территория), супер-эффективность для ранжирования и индекс Мальмквиста (динамика производительности). Нежелательные выходы учитываются по схеме «undesirable as input».

Drivers. Random Forest предсказывает интегральный индекс; вклад факторов - важность по Джини, перестановочная важность и SHAP.

2.4 Входы и выходы DEA (пример, адаптируемый под данные)

Заданы в config.py по логике оценки эффективности АПК и легко редактируются под состав вашего датасета:

Inputs: площадь сельхозугодий; работающее население; число хозяйств (организационный капитал).

Desirable outputs: поголовье скота; NDVI как прокси биопродуктивности угодий.

Undesirable outputs: застройка; ночное освещение (NLT).

2.5 Порядок запуска

1) Установить зависимости: `pip install -r requirements.txt` (обязательны pandas, numpy, openpyxl; для драйверов - scikit-learn; shap и matplotlib - опционально; scipy НЕ требуется).

2) Поместить свой датасет как data.xlsx рядом с кодом (или указать путь в config.py); при необходимости поправить имена колонок и входы/выходы DEA в config.py.

3) Выполнить весь пайплайн: `python run_all.py` (базовый) или `python pipeline_full.py` (расширенный, с DL и ансамблями)

2.6 Результирующие файлы

После запуска результаты сохраняются в каталог outputs/:

File	Содержимое
correlation_pearson / spearman / kendall.xlsx	матрицы парных корреляций
strong_correlations.xlsx	сильные связи $ r \geq 0.70$
integral_index.xlsx	интегральный индекс и класс по каждой территории-году
integral_weights.xlsx	веса показателей (энтропийный метод)
dea_efficiency.xlsx	DEA: CCR и супер-эффективность по годам
dea_malmquist.xlsx	индекс Мальмквиста (динамика производительности)
ml_metrics.xlsx	качество модели (R^2 , RMSE, MAE, CV)
ml_feature_importance.xlsx / ml_shap_importance.xlsx	важности и SHAP-вклад факторов

Раздел драйверов (Random Forest + SHAP) формируется при наличии scikit-learn / shap.

2.7 Полный текст скрипта

Ниже приведён полный код всех модулей репозитория.

2.7.1 Модуль *config.py*

```
# -*- coding: utf-8 -*-  
"""
```

Конфигурация анализа АПК по панельным данным «округ-год».

Инструмент обезличен: подставьте СВОЙ датасет и при необходимости поправьте названия колонок ниже. Все колонки ищутся по ПОДСТРОКЕ (регистр и хвостовые пробелы игнорируются), поэтому точные полные названия указывать не обязательно.

Здесь задаются: путь к данным, идентификаторы панели, тематические группы показателей, входы/выходы для DEA, целевая переменная и параметры моделей.

```
"""  
  
from pathlib import Path  
  
# -----  
# -----  
# Данные (подставьте свои)  
# -----  
# -----  
# Поместите файл с данными рядом с кодом  
под именем data.xlsx  
# либо укажите здесь свой путь. SHEET_NAME  
= 0 - первый лист книги.  
BASE_DIR = Path(__file__).resolve().parent
```

```

DATA_PATH = BASE_DIR / "data.xlsx"
SHEET_NAME = 0
OUTPUT_DIR = BASE_DIR / "outputs"

# -----
# Идентификаторы панели (имена колонок в
вашем датасете)
# -----
ID_NAME = "Name"      # территория
(например, сельский округ)
ID_YEAR = "Year_"     # год

# -----
# Тематические группы показателей (по
подстроке) - пример, адаптируйте
# -----
GROUPS = {
    "Дистанционное зондирование": ["NDVI",
    "EVI", "NDWI", "NDTI", "NDMI", "NDBI",
    "MNDWI", "NLT"],
    "Землепользование (LULC)": ["Водные
объекты", "Городская застройка", "Леса",
                                "Пастбища
и травы", "Пашни"],
    "Демография": ["численность населения",
    "число родившихся", "число умерших",
                                "число прибывших",
    "число выбывших"],
    "Социум": ["детских садов", "детей
дошкольного", "типовых школ", "школьников",
                                "преподавателей",
    "работающее население", "Врачей", "Медицинских
работников"],

```

```

        "медицинских учреждений",
"Общее количество домов", "центральной
питьевой",
        "сетевым интернетом",
"библиотек", "аптек"],
        "Экономика АПК": ["заработная плата",
"Площадь сельхозугодий", "пашни", "пастбища",
        "рогатого скота",
"Овец и коз", "Лошадей", "Свиней", "Птиц",
        "крестьянских
хозяйств", "ограниченной ответственностью",
        "акционерное
общество", "кооперативов", "торговые
заведения"],
    }

# -----
-----

# DEA: входы и выходы (логика оценки
эффективности АПК) - пример, адаптируйте
# -----
-----

# Входы (ресурсы): земля, труд,
организационный капитал.
DEA_INPUTS = [
    "Площадь сельхозугодий",
    "работающее население",
    "крестьянских хозяйств",
    "ограниченной ответственностью",
    "кооперативов",
]

# Желательные выходы (результаты
производства).
DEA_DESIRABLE = [
    "рогатого скота",
    "Овец и коз",
    "Лошадей",
    "NDVI",

```

```

]
# Нежелательные выходы (антропогенная
нагрузка) - схема "undesirable as input".
DEA_UNDESIRABLE = [
    "Городская застройка",
    "NLT",
]

# -----
# Машинное обучение (драйверы интегральной
оценки)
# -----
ML_TARGET = "Integral_Index"
ML_RANDOM_STATE = 42
ML_TEST_SIZE = 0.25
RF_PARAMS = dict(n_estimators=400,
max_depth=None, min_samples_leaf=3,

random_state=ML_RANDOM_STATE, n_jobs=-1)

# -----
# Интегральный индекс
# -----
INTEGRAL_WEIGHTING = "entropy" #
"entropy" или "equal"
STRONG_CORR_THRESHOLD = 0.70

```

2.7.2 Модуль *apk_io.py*

```

# -*- coding: utf-8 -*-
"""Загрузка данных и сопоставление колонок
по подстроке."""
import unicodedata

```

```

import pandas as pd
import config

def _norm(s: str) -> str:
    """Нормализация строки для нечёткого
сравнения колонок."""
    s = str(s).strip().lower().replace("ё",
    "е")
    return unicodedata.normalize("NFKC", s)

def load_dataset() -> pd.DataFrame:
    """Читает входной датасет (панель
«территория-год»)."""
    df = pd.read_excel(config.DATA_PATH,
sheet_name=config.SHEET_NAME)
    df = df.dropna(how="all")
    if config.ID_YEAR in df.columns:
        df[config.ID_YEAR] =
pd.to_numeric(df[config.ID_YEAR],
errors="coerce")
        msg = f"[load] {df.shape[0]} записей x
{df.shape[1]} колонок"
        if config.ID_NAME in df.columns:
            msg += f" | территорий:
{df[config.ID_NAME].nunique()}"
            if config.ID_YEAR in df.columns and
df[config.ID_YEAR].notna().any():
                msg += f" | годы:
{int(df[config.ID_YEAR].min())}-
{int(df[config.ID_YEAR].max())}"
            print(msg)
        return df

def find_col(df: pd.DataFrame, substring:
str):

```

```

        """Имя первой колонки, содержащей
подстроку (нечётко)."""
        target = _norm(substring)
        for col in df.columns:
            if target in _norm(col):
                return col
        return None

def find_cols(df: pd.DataFrame,
substrings):
    """Список найденных колонок (пропуски
игнорируются)."""
    out = []
    for s in substrings:
        c = find_col(df, s)
        if c is not None and c not in out:
            out.append(c)
    return out

def numeric_columns(df: pd.DataFrame):
    """Числовые колонки без идентификаторов
панели."""
    ids = {config.ID_NAME, config.ID_YEAR}
    return [c for c in df.columns
            if c not in ids and
pd.api.types.is_numeric_dtype(df[c])]

def ensure_outdir():
    config.OUTPUT_DIR.mkdir(parents=True,
exist_ok=True)
    return config.OUTPUT_DIR

```

2.7.3 Модуль *preprocessing.py*

```
# -*- coding: utf-8 -*-
"""Предобработка и нормировка показателей
(без внешних зависимостей)."""
import numpy as np
import pandas as pd

def minmax(s: pd.Series, inverse: bool =
False) -> pd.Series:
    """Min-Max нормировка в [0,1].
inverse=True для негативных показателей."""
    s = pd.to_numeric(s, errors="coerce")
    lo, hi = s.min(), s.max()
    if pd.isna(lo) or hi == lo:
        return pd.Series(0.5,
index=s.index)
    z = (s - lo) / (hi - lo)
    return (1 - z if inverse else
z).fillna(0.5)

def robust(s: pd.Series, inverse: bool =
False) -> pd.Series:
    """Робастная нормировка по 1-му и 99-му
процентильям (устойчива к выбросам)."""
    s = pd.to_numeric(s, errors="coerce")
    q1, q99 = s.quantile(0.01),
s.quantile(0.99)
    if q99 == q1:
        return pd.Series(0.5,
index=s.index)
    z = ((s - q1) / (q99 - q1)).clip(0, 1)
    return (1 - z if inverse else
z).fillna(0.5)
```

```

def zscore(s: pd.Series) -> pd.Series:
    """Стандартизация (StandardScaler):
    среднее 0, дисперсия 1."""
    s = pd.to_numeric(s, errors="coerce")
    mu, sd = s.mean(), s.std(ddof=0)
    if sd == 0 or pd.isna(sd):
        return pd.Series(0.0,
index=s.index)
    return ((s - mu) / sd).fillna(0.0)

def normalize_frame(df: pd.DataFrame, cols,
method: str = "minmax",
inverse_cols=None) ->
pd.DataFrame:
    """Нормирует выбранные колонки
    указанным методом."""
    inverse_cols = set(inverse_cols or [])
    func = {"minmax": minmax, "robust":
robust, "zscore": zscore}[method]
    out = pd.DataFrame(index=df.index)
    for c in cols:
        if method == "zscore":
            out[c] = func(df[c])
        else:
            out[c] = func(df[c], inverse=(c
in inverse_cols))
    return out

```

2.7.4 Модуль correlation.py

```

# -*- coding: utf-8 -*-
"""Корреляционный анализ полного массива
(Пирсон / Спирмен / Кендалл)."""
import itertools
import pandas as pd
import config

```

```

import apk_io

def run(df: pd.DataFrame, outdir):
    cols = apk_io.numeric_columns(df)
    data = df[cols].apply(pd.to_numeric,
errors="coerce")
    print(f"[corr] числовых показателей:
{len(cols)} | "
        f"map: {len(cols) * (len(cols) -
1) // 2}")

    methods = ["pearson", "spearman",
"kendall"]
    strong_rows = []
    for m in methods:
        try:
            cm = data.corr(method=m)
        except Exception as e:
            print(f"[corr] {m}: пропущен
({e}). Нужен scipy для kendall.")
            continue
        cm.to_excel(outdir /
f"correlation_{m}.xlsx")
        # сильные связи |r| > порога
        for a, b in
itertools.combinations(cols, 2):
            r = cm.loc[a, b]
            if pd.notna(r) and abs(r) >=
config.STRONG_CORR_THRESHOLD:

            strong_rows.append({"method": m, "var_1": a,
"var_2": b, "r": round(float(r), 4)})

    strong =
pd.DataFrame(strong_rows).sort_values(
    ["method", "r"], key=lambda s:
s.abs() if s.name == "r" else s,

```

```

        ascending=[True, False]) if
strong_rows else pd.DataFrame(
        columns=["method", "var_1",
"var_2", "r"])
        strong.to_excel(outdir /
"strong_correlations.xlsx", index=False)
        print(f"[corr] сильных связей
|r|>={config.STRONG_CORR_THRESHOLD}:
{len(strong)}")
        return strong

```

2.7.5 Модуль *integral_index.py*

```

# -*- coding: utf-8 -*-
"""

```

Интегральная оценка развития АПК.

Шаги: нормировка показателей -> объективное взвешивание (энтропийный метод)

-> агрегирование в сводный индекс -> классификация по квантилям.

Негативные показатели (нагрузка) нормируются инверсно.

```

"""

```

```

import numpy as np
import pandas as pd
import config
import apk_io
import preprocessing as pp

```

```

NEGATIVE_HINTS = ["Городская застройка",
"NDBI", "NDTI", "NLT",
                  "число умерших", "число
выбывших"]

```

```

def entropy_weights(norm: pd.DataFrame) ->
pd.Series:
    """Объективные веса по энтропийному
методу (нормированная [0,1] матрица)."""
    P = norm.clip(lower=1e-12)
    P = P.div(P.sum(axis=0), axis=1)
    k = 1.0 / np.log(len(P))
    E = -k * (P * np.log(P)).sum(axis=0)
    d = 1.0 - E
    if d.sum() == 0:
        return pd.Series(1.0 /
norm.shape[1], index=norm.columns)
    return d / d.sum()

def run(df: pd.DataFrame, outdir):
    cols = apk_io.numeric_columns(df)
    neg = set(apk_io.find_cols(df,
NEGATIVE_HINTS))
    # робастная нормировка устойчивее к
скошенности счётных показателей
    norm = pp.normalize_frame(df, cols,
method="robust", inverse_cols=neg)

    if config.INTEGRAL_WEIGHTING ==
"entropy":
        w = entropy_weights(norm)
    else:
        w = pd.Series(1.0 / len(cols),
index=cols)

    index = (norm * w).sum(axis=1)
    out = df[[config.ID_NAME,
config.ID_YEAR]].copy()
    out[config.ML_TARGET] = index.round(4)
    out["Class"] = pd.qcut(index, 4,
labels=["низкий", "ниже среднего",

```

```

"выше среднего", "высокий"]])
    out.to_excel(outdir /
"integral_index.xlsx", index=False)

w.sort_values(ascending=False).round(4).to_frame("weight").to_excel(
    outdir / "integral_weights.xlsx")
    print(f"[index] интегральный индекс
рассчитан для {len(out)} наблюдений | "
        f"метод весов:
{config.INTEGRAL_WEIGHTING}")
    merged = df.merge(out[[config.ID_NAME,
config.ID_YEAR, config.ML_TARGET]],
                    on=[config.ID_NAME,
config.ID_YEAR], how="left")
    return merged

```

2.7.6 Модуль *dea_lp.py*

```

# -*- coding: utf-8 -*-
"""

```

DEA на чистом NumPy: собственный двухфазный симплекс-метод (без scipy).

Реализовано:

```

* linprog_min          - решатель ЛП в
стандартной форме;
* dea_scores           - входо-
ориентированная CCR и супер-эффективность;
* malmquist            - индекс Мальмквиста
(динамика производительности).

```

Проверено на аналитических примерах (см. блок `__main__`).

```

"""

```

```

import warnings

```

```

import numpy as np

warnings.filterwarnings("ignore",
category=RuntimeWarning)

def linprog_min(c, A_ub=None, b_ub=None,
tol=1e-9, maxit=10000):
    """min c@x при A_ub@x <= b_ub, x >= 0.
    Возвращает (x, opt) или (None, None)."""
    c = np.asarray(c, float)
    n = c.size
    A_ub = np.asarray(A_ub, float) if A_ub
is not None else np.zeros((0, n))
    b_ub = np.asarray(b_ub, float) if b_ub
is not None else np.zeros(0)
    m = A_ub.shape[0]
    total = n + m
    A = np.hstack([A_ub, np.eye(m)])
    b = b_ub.copy()
    for i in range(m):
        if b[i] < 0:
            A[i] *= -1.0
            b[i] *= -1.0

    def simplex(cost, A, b, basis):
        A, b, cost, basis = A.copy(),
b.copy(), cost.copy(), basis[:]
        for _ in range(maxit):
            z = cost[basis] @ A - cost
            j = int(np.argmax(z))
            if z[j] <= tol:
                break
            col = A[:, j]
            if np.all(col <= tol):
                return None, None, None
            ratios = np.where(col > tol, b
/ col, np.inf)

```

```

        i = int(np.argmin(ratios))
        piv = A[i, j]
        A[i] /= piv
        b[i] /= piv
        piv_row = A[i].copy()
        piv_b = b[i]
        for k in range(A.shape[0]):
            if k != i:
                f = A[k, j]
                A[k] -= f * piv_row
                b[k] -= f * piv_b
        basis[i] = j
    return A, b, basis

A1 = np.hstack([A, np.eye(m)])
c1 = np.zeros(total + m)
c1[total:] = 1.0
basis = list(range(total, total + m))
A1, b1, basis = simplex(c1, A1, b,
basis)
    if A1 is None or c1[basis] @ b1 > 1e-6:
        return None, None
    c2 = np.zeros(total + m)
    c2[:n] = c
    c2[total:] = 1e9
    A2, b2, basis = simplex(c2, A1, b1,
basis)
    if A2 is None:
        return None, None
    x = np.zeros(total + m)
    for i, bi in enumerate(basis):
        x[bi] = b2[i]
    return x[:n], float(c @ x[:n])

def _solve_ccr(X, Y, o,
reference_mask=None):

```

```

        """Входо-ориентированная CCR-
эффективность для DMU o."""
        n, ni = X.shape
        no = Y.shape[1]
        cols = np.ones(n, bool) if
reference_mask is None else reference_mask
        Xr, Yr = X[cols], Y[cols]
        nr = Xr.shape[0]
        c = np.zeros(nr + 1)
        c[0] = 1.0
        A, b = [], []
        for i in range(ni):
            row = np.zeros(nr + 1)
            row[0] = -X[o, i]
            row[1:] = Xr[:, i]
            A.append(row); b.append(0.0)
        for r in range(no):
            row = np.zeros(nr + 1)
            row[1:] = -Yr[:, r]
            A.append(row); b.append(-Y[o, r])
        _, opt = linprog_min(c, A_ub=A, b_ub=b)
        return opt

```

```

def _prepare(inputs, desirable,
undesirable):
    """X = входы (+ нежелательные как
входы), Y = желательные. Всё сдвигается >
0."""
    X_parts = [np.asarray(inputs, float)]
    if undesirable is not None and
np.size(undesirable):
        X_parts.append(np.asarray(undesirable, float))
    X = np.hstack(X_parts)
    Y = np.asarray(desirable, float)

    def positivize(M):

```

```

        M = M.copy()
        for j in range(M.shape[1]):
            col = M[:, j]
            col = np.where(np.isnan(col),
np.nanmin(col), col)
            mn = np.nanmin(col)
            if mn <= 0:
                col = col - mn + 1e-6
            M[:, j] = col
        return M

    return positivize(X), positivize(Y)

```

```

def dea_scores(inputs, desirable,
undesirable=None, super_eff=False):
    """Вектор эффективностей DMU (CCR или
супер-эффективность)."""
    X, Y = _prepare(inputs, desirable,
undesirable)
    n = X.shape[0]
    out = np.full(n, np.nan)
    for o in range(n):
        if super_eff:
            mask = np.ones(n, bool);
mask[o] = False
            out[o] = _solve_ccr(X, Y, o,
reference_mask=mask)
        else:
            out[o] = _solve_ccr(X, Y, o)
    return out

```

```

def malmquist(X_t, Yd_t, X_t1, Yd_t1,
Yu_t=None, Yu_t1=None):
    """Индекс Мальмквиста между t и t+1.
M>1 - рост производительности.

```

Периоды приводятся к положительным значениям ОБЩИМ сдвигом по столбцам (единый масштаб), иначе межпериодные расстояния несопоставимы.

```

"""
def stackX(inp, und):
    parts = [np.asarray(inp, float)]
    if und is not None and
np.size(und):
        parts.append(np.asarray(und,
float))
    return np.hstack(parts)

def joint_pos(A, B):
    A = A.copy(); B = B.copy()
    for j in range(A.shape[1]):
        both = np.concatenate([A[:, j],
B[:, j]])
        mn = np.nanmin(both)
        A[:, j] =
np.where(np.isnan(A[:, j]), mn, A[:, j])
        B[:, j] =
np.where(np.isnan(B[:, j]), mn, B[:, j])
        if mn <= 0:
            A[:, j] = A[:, j] - mn +
1e-6
            B[:, j] = B[:, j] - mn +
1e-6
    return A, B

Xt, Xt1 = joint_pos(stackX(X_t, Yu_t),
stackX(X_t1, Yu_t1))
Yt, Yt1 = joint_pos(np.asarray(Yd_t,
float), np.asarray(Yd_t1, float))
n = Xt.shape[0]
M = np.full(n, np.nan)

def dist(Xref, Yref, x_o, y_o):

```

```

        nr = Xref.shape[0]
        c = np.zeros(nr + 1); c[0] = 1.0
        A, b = [], []
        for i in range(Xref.shape[1]):
            row = np.zeros(nr + 1); row[0]
= -x_o[i]; row[1:] = Xref[:, i]
            A.append(row); b.append(0.0)
            for r in range(Yref.shape[1]):
                row = np.zeros(nr + 1); row[1:]
= -Yref[:, r]
                A.append(row); b.append(-
y_o[r])
        _, opt = linprog_min(c, A_ub=A,
b_ub=b)
        return opt

    eps = 0.02
    for o in range(n):
        d_t_t = dist(Xt, Yt, Xt[o], Yt[o])
        d_t_t1 = dist(Xt, Yt, Xt1[o],
Yt1[o])
        d_t1_t1 = dist(Xt1, Yt1, Xt1[o],
Yt1[o])
        d_t1_t = dist(Xt1, Yt1, Xt[o],
Yt[o])
        vals = [d_t_t, d_t_t1, d_t1_t1,
d_t1_t]
        if any(v is None or not
np.isfinite(v) or v < eps for v in vals):
            M[o] = np.nan
            continue
        m = float(np.sqrt((d_t_t1 / d_t_t)
* (d_t1_t1 / d_t1_t)))
        # фильтр правдоподобия: вне [0.05;
20] значение численно вырождено
        # (близкая к нулю собственная
дистанция / аномальный скачок данных)

```

```

        M[o] = m if 0.05 <= m <= 20.0 else
np.nan
    return M

if __name__ == "__main__":
    X = np.array([[1.], [2.], [4.], [3.]])
Y = np.array([[1.], [1.], [3.], [2.]])
    e = dea_scores(X, Y)
    ratios = Y[:, 0] / X[:, 0]
    expected = ratios / ratios.max()
    print("CCR:", np.round(e, 4), "|
ожидание:", np.round(expected, 4),
        "| совпало:", np.allclose(e,
expected, atol=1e-4))
    es = dea_scores(X, Y, super_eff=True)
    print("Super:", np.round(es, 4))

```

2.7.7 Модуль dea_analysis.py

```

# -*- coding: utf-8 -*-
"""
Оценка эффективности АПК методом DEA по
годам.

Для каждого года территории рассматриваются
как DMU (единицы оценки). Рассчитываются:
    * CCR-эффективность (входо-
ориентированная);
    * супер-эффективность (для ранжирования
эффективных округов);
    а также индекс Мальмквиста год-к-году
(изменение производительности).
"""
import numpy as np
import pandas as pd
import config

```

```

import apk_io
import dea_lp

def _matrices(block: pd.DataFrame, in_cols,
des_cols, und_cols):
    X = block[in_cols].apply(pd.to_numeric,
errors="coerce").to_numpy()
    Yd =
block[des_cols].apply(pd.to_numeric,
errors="coerce").to_numpy()
    Yu =
block[und_cols].apply(pd.to_numeric,
errors="coerce").to_numpy() if und_cols else
None
    return X, Yd, Yu

def run(df: pd.DataFrame, outdir):
    in_cols = apk_io.find_cols(df,
config.DEA_INPUTS)
    des_cols = apk_io.find_cols(df,
config.DEA_DESIRABLE)
    und_cols = apk_io.find_cols(df,
config.DEA_UNDESIRABLE)
    print(f"[dea] входы: {len(in_cols)} |
желательные выходы: {len(des_cols)} | "
        f"нежелательные:
{len(und_cols)}")

    results = []
    years = sorted(int(y) for y in
df[config.ID_YEAR].dropna().unique())
    blocks = {}
    for y in years:
        block = df[df[config.ID_YEAR] ==
y].dropna(subset=in_cols + des_cols).copy()
        if len(block) < 3:

```

```

        continue
        blocks[y] = block
        X, Yd, Yu = _matrices(block,
in_cols, des_cols, und_cols)
        eff = dea_lp.dea_scores(X, Yd, Yu,
super_eff=False)
        sup = dea_lp.dea_scores(X, Yd, Yu,
super_eff=True)
        for name, e, s in
zip(block[config.ID_NAME].values, eff, sup):
            results.append({"Name": name,
"Year": y,
                                "DEA_CCR":
round(float(e), 4) if e is not None else
np.nan,
                                "DEA_Super":
round(float(s), 4) if s is not None else
np.nan})
        res = pd.DataFrame(results)
        res.to_excel(outdir /
"dea_efficiency.xlsx", index=False)

# Индекс Мальмквиста между соседними
годами по общим округам
malm_rows = []
yrs = sorted(blocks)
for t, t1 in zip(yrs, yrs[1:]):
    b0, b1 = blocks[t], blocks[t1]
    common =
sorted(set(b0[config.ID_NAME]) &
set(b1[config.ID_NAME]))
    if not common:
        continue
    b0c =
b0.set_index(config.ID_NAME).loc[common]
    b1c =
b1.set_index(config.ID_NAME).loc[common]

```

```

        X0, Yd0, Yu0 = _matrices(b0c,
in_cols, des_cols, und_cols)
        X1, Yd1, Yu1 = _matrices(b1c,
in_cols, des_cols, und_cols)
        M = dea_lp.malmquist(X0, Yd0, X1,
Yd1, Yu0, Yu1)
        for name, m in zip(common, M):
            malm_rows.append({"Name": name,
"Period": f"{t}-{t1}",
                                "Malmquist":
round(float(m), 4) if np.isfinite(m) else
np.nan})
        pd.DataFrame(malm_rows).to_excel(outdir
/ "dea_malmquist.xlsx", index=False)
        print(f"[dea] рассчитано DMU-лет:
{len(res)} | пар Мальмквиста:
{len(malm_rows)}")
        return res

```

2.7.8 Модуль *ml_drivers.py*

```

# -*- coding: utf-8 -*-
"""
Анализ факторов-драйверов интегральной
оценки.

Модель: Random Forest (sklearn).
Интерпретация: важность признаков,
перестановочная важность и SHAP (если
установлен пакет shap).
Качество оценивается на отложенной выборке
(R2, RMSE, MAE) и кросс-валидацией.

Требует: scikit-learn (обязательно), shap
(опционально).
"""
import numpy as np

```

```

import pandas as pd
import config
import apk_io

def run(df: pd.DataFrame, outdir):
    try:
        from sklearn.ensemble import
RandomForestRegressor
        from sklearn.model_selection import
train_test_split, cross_val_score
        from sklearn.inspection import
permutation_importance
        from sklearn.metrics import
r2_score, mean_squared_error,
mean_absolute_error
    except ImportError:
        print("[ml] scikit-learn не
установлен - раздел пропущен. "
              "Установите: pip install
scikit-learn")
        return None

    target = config.ML_TARGET
    if target not in df.columns:
        print(f"[ml] нет целевой '{target}'
- сначала рассчитайте интегральный индекс.")
        return None

    feats = [c for c in
apk_io.numeric_columns(df) if c != target]
    data = df[feats +
[target]].apply(pd.to_numeric,
errors="coerce").dropna()
    X, y = data[feats], data[target]

    Xtr, Xte, ytr, yte = train_test_split(

```

```

        X, y,
test_size=config.ML_TEST_SIZE,
random_state=config.ML_RANDOM_STATE)
    model =
RandomForestRegressor(**config.RF_PARAMS)
    model.fit(Xtr, ytr)
    pred = model.predict(Xte)

    metrics = {
        "R2_test":
round(float(r2_score(yte, pred)), 4),
        "RMSE_test":
round(float(np.sqrt(mean_squared_error(yte,
pred))), 4),
        "MAE_test":
round(float(mean_absolute_error(yte, pred)),
4),
        "R2_CV5_mean":
round(float(cross_val_score(
            model, X, y, cv=5,
scoring="r2").mean()), 4),
    }

pd.Series(metrics).to_frame("value").to_excel(
outdir / "ml_metrics.xlsx")
    print(f"[ml] RandomForest | {metrics}")

    # важность признаков (Gini) +
перестановочная важность
    imp = pd.DataFrame({
        "feature": feats,
        "importance_gini":
model.feature_importances_,
    })
    perm = permutation_importance(model,
Xte, yte, n_repeats=10,

random_state=config.ML_RANDOM_STATE)

```

```

        imp["importance_perm"] =
perm.importances_mean
        imp =
imp.sort_values("importance_gini",
ascending=False)
        imp.round(5).to_excel(outdir /
"ml_feature_importance.xlsx", index=False)

        # SHAP (если доступен)
        try:
            import shap
            explainer =
shap.TreeExplainer(model)
            sv = explainer.shap_values(Xte)
            shap_imp = pd.DataFrame({
                "feature": feats,
                "mean_abs_shap":
np.abs(sv).mean(axis=0),
            }).sort_values("mean_abs_shap",
ascending=False)
            shap_imp.round(5).to_excel(outdir /
"ml_shap_importance.xlsx", index=False)
            print(f"[ml] SHAP: топ-драйвер -
{shap_imp.iloc[0]['feature']}")
        except ImportError:
            print("[ml] shap не установлен -
SHAP пропущен (pip install shap).")
        return imp

```

2.7.9 Модуль *data_gsheets.py*

```

# -*- coding: utf-8 -*-
"""
Загрузка панели «территория-год» напрямую
из облачной таблицы (источник [1]).

```

Модуль является альтернативой чтению локального data.xlsx (apk_io.py):
данные выгружаются частями через CSV-интерфейс Google Visualization API,
десятичные запятые приводятся к точкам,
схема (число колонок и строк)
проверяется до передачи в конвейер.
Результат кешируется в data.xlsx,
поэтому повторные запуски не требуют подключения к сети.

"""

```
import io
import time
import urllib.parse
import urllib.request
```

```
import numpy as np
import pandas as pd
```

```
import config
```

```
# Идентификатор книги (см. позицию [1]
списка литературы).
```

```
SPREADSHEET_ID =
"1EmZ5U6Akl2HOMYKerQJYHOvDfiomd-nx"
```

```
# Буквенные группы колонок: широкие блоки
не выгружаются одним запросом,
```

```
# чтобы не упереться в ограничение на
размер ответа.
```

```
COLUMN_GROUPS = [
    "A,B,C,D,E",          # территория,
год, LULC-площади
    "F,G,H,I,J,K",        # пастбища,
пашни, NDVI, EVI, NDWI, NDTI
    "L,M,N,O",            # NDMI, NDBI,
MNDWI, NLT
```

```

        "P,Q,R,S,T",          # демография
        "U,V,W,X,Y,Z",      # образование,
занятость
        "AA,AB,AC,AD,AE,AF", #
здоровоохранение, жильё, инфраструктура
        "AG,AH,AI,AJ,AK,AL", # библиотеки,
аптеки, зарплата, угодья
        "AM,AN,AO,AP,AQ",    # поголовье скота
и птицы
        "AR,AS,AT,AU,AV",    # формы
хозяйствования, торговля
    ]

```

```

EXPECTED_ROWS = 1260    # 84 округа x 15 лет
EXPECTED_COLS = 48      # 46 показателей + 2
служебных поля

```

```

def _gviz_url(columns: str, limit: int = 0,
offset: int = 0) -> str:
    """Собрать URL запроса к Google
Visualization API (ответ - CSV)."""
    query = "select " + columns
    if limit:
        query += f" limit {limit}"
    if offset:
        query += f" offset {offset}"
    return (

f"https://docs.google.com/spreadsheets/d/{SPREADSHEET_ID}"
        f"/gviz/tq?tqx=out:csv&tq=" +
urllib.parse.quote(query)
    )

```

```

def _fetch_csv(url: str, retries: int = 3,
pause: float = 2.0) -> str:

```

```

        """Скачать CSV с повторами при сетевых
сбоях."""
        last_err = None
        for attempt in range(retries):
            try:
                with
urllib.request.urlopen(url, timeout=60) as
resp:
                    return
resp.read().decode("utf-8")
            except Exception as err:          #
noqa: BLE001 - повторяем
                last_err = err
                time.sleep(pause * (attempt +
1))
            raise RuntimeError(f"Не удалось
получить данные: {last_err}")

def _to_number(value):
    """'1 234,56' -> 1234.56; пустые строки
-> NaN; текст остаётся текстом."""
    if isinstance(value, str):
        cleaned = value.replace("\xa0",
"").replace(" ", "").replace(",", ".")
        try:
            return float(cleaned)
        except ValueError:
            return value
    return value

def load_from_cloud(verbose: bool = True) -
> pd.DataFrame:
    """Выгрузить все группы колонок и
склеить их в одну таблицу."""
    blocks = []
    for group in COLUMN_GROUPS:

```

```

        raw = _fetch_csv(_gviz_url(group))
        block =
pd.read_csv(io.StringIO(raw), dtype=str)
        if verbose:
            print(f" колонки {group}:
{block.shape[0]} строк")

blocks.append(block.reset_index(drop=True))
df = pd.concat(blocks, axis=1)

        if df.shape[0] != EXPECTED_ROWS or
df.shape[1] != EXPECTED_COLS:
            raise ValueError(
                f"Схема не совпала: получено
{df.shape}, "
                f"ожидалось ({EXPECTED_ROWS},
{EXPECTED_COLS})"
            )

        numeric_cols = [c for c in df.columns
if c != config.ID_NAME]
        for col in numeric_cols:
            df[col] = df[col].map(_to_number)
            df[col] = pd.to_numeric(df[col],
errors="coerce")

        df[config.ID_YEAR] =
df[config.ID_YEAR].astype(int)
        return df

def load_or_cache() -> pd.DataFrame:
    """Вернуть кеш data.xlsx, при его
отсутствии - скачать и сохранить."""
    if config.DATA_PATH.exists():
        return
pd.read_excel(config.DATA_PATH,
sheet_name=config.SHEET_NAME)

```

```

        df = load_from_cloud()
        df.to_excel(config.DATA_PATH,
index=False)
        print("Кеш сохранён:",
config.DATA_PATH)
        return df

```

```

if __name__ == "__main__":
    frame = load_or_cache()
    print(frame.shape)
    print(frame.head())

```

2.7.10 Модуль *dl_models.py*

```

# -*- coding: utf-8 -*-
"""
Модели глубокого обучения (глава 5):
SimpleTF, DeepTF, ResidualNet,
LSTM и BiLSTM. Целевая переменная - DEA-
индекс округа (или интегральный
индекс, см. config.ML_TARGET).

Протокол валидации единый с ML-блоком:
разбиение 75/25 с фиксированным
random_state, группировка по округам
(GroupKFold) против «утечки» между
связанными наблюдениями; для рекуррентных
моделей - разбиение по времени
(ранние годы в обучение, поздние - в тест).
"""

import numpy as np
import pandas as pd

from sklearn.model_selection import
GroupShuffleSplit

```

```

    from sklearn.preprocessing import
StandardScaler
    from sklearn.metrics import
mean_absolute_error, mean_squared_error,
r2_score

    import tensorflow as tf
    from tensorflow.keras import layers,
models, callbacks

    import config

    SEED = config.ML_RANDOM_STATE
    tf.random.set_seed(SEED)
    np.random.seed(SEED)

    EPOCHS = 300
    BATCH = 64
    PATIENCE = 30

# -----
# -----
# Архитектуры полносвязных сетей
# -----
# -----

def build_simple_tf(n_features: int) ->
models.Model:
    """«Компактная» сеть: 2 скрытых слоя
умеренной ширины + Dropout."""
    net =
models.Sequential(name="SimpleTF")

net.add(layers.Input(shape=(n_features,)))
    net.add(layers.Dense(128,
activation="relu"))
    net.add(layers.BatchNormalization())
    net.add(layers.Dropout(0.20))

```

```

        net.add(layers.Dense(64,
activation="relu"))
        net.add(layers.Dropout(0.10))
        net.add(layers.Dense(1,
activation="linear"))

net.compile(optimizer=tf.keras.optimizers.Adam
(1e-3), loss="mse",
            metrics=["mae"])
return net

```

```

def build_deep_tf(n_features: int) ->
models.Model:
    """«Глубокая» сеть: 5 скрытых слоёв с
сужением 512 -> 32."""
    net = models.Sequential(name="DeepTF")

net.add(layers.Input(shape=(n_features,)))
    for width in (512, 256, 128, 64, 32):
        net.add(layers.Dense(width,
activation="relu"))

net.add(layers.BatchNormalization())
        net.add(layers.Dropout(0.25))
        net.add(layers.Dense(1,
activation="linear"))

net.compile(optimizer=tf.keras.optimizers.Adam
(5e-4), loss="mse",
            metrics=["mae"])
return net

```

```

def build_residual_net(n_features: int,
blocks: int = 3) -> models.Model:
    """Полносвязная сеть с остаточными
связями (skip connections)."""

```

```

        inp = layers.Input(shape=(n_features,))
        x = layers.Dense(128,
activation="relu")(inp)
        for _ in range(blocks):
            skip = x
            x = layers.Dense(128,
activation="relu")(x)
            x = layers.BatchNormalization()(x)
            x = layers.Dropout(0.15)(x)
            x = layers.Dense(128,
activation="linear")(x)
            x = layers.Add()([x, skip])
            x = layers.Activation("relu")(x)
            out = layers.Dense(1,
activation="linear")(x)
        net = models.Model(inp, out,
name="ResidualNet")

net.compile(optimizer=tf.keras.optimizers.Adam
(1e-3), loss="mse",
            metrics=["mae"])
    return net

```

```

# -----
# Рекуррентные модели: последовательности
по каждому округу
# -----
def build_lstm(window: int, n_features:
int,
                bidirectional: bool = False)
-> models.Model:
    """LSTM/BiLSTM над окном из `window`
лет наблюдений округа."""
    rnn = layers.LSTM(64,
return_sequences=False)

```

```

        if bidirectional:
            rnn = layers.Bidirectional(rnn)
        net = models.Sequential(name="BiLSTM"
if bidirectional else "LSTM")
        net.add(layers.Input(shape=(window,
n_features)))
        net.add(rnn)
        net.add(layers.Dropout(0.20))
        net.add(layers.Dense(32,
activation="relu"))
        net.add(layers.Dense(1,
activation="linear"))

net.compile(optimizer=tf.keras.optimizers.Adam
(1e-3), loss="mse",
            metrics=["mae"])
return net

```

```

def make_sequences(df: pd.DataFrame,
features: list, target: str,
                    window: int = 3):
    """Скользящие окна лет внутри каждого
округа (без смещения округов)."""
    xs, ys, years = [], [], []
    for _, sub in
df.sort_values(config.ID_YEAR).groupby(config.
ID_NAME):
        values =
sub[features].to_numpy(dtype="float32")
        targets =
sub[target].to_numpy(dtype="float32")
        yrs =
sub[config.ID_YEAR].to_numpy()
        for k in range(window, len(sub)):
            xs.append(values[k - window:k])
            ys.append(targets[k])
            years.append(yrs[k])

```

```

        return np.stack(xs), np.array(ys),
        np.array(years)

# -----
# Обучение и сводка метрик
# -----
def _metrics(y_true, y_pred) -> dict:
    mse = mean_squared_error(y_true,
y_pred)
    return {
        "R2": round(r2_score(y_true,
y_pred), 3),
        "RMSE": round(float(np.sqrt(mse)),
3),
        "MAE":
round(mean_absolute_error(y_true, y_pred), 3),
        "MSE": round(mse, 4),
    }

def run(df: pd.DataFrame, outdir, features:
list, target: str) -> dict:
    """Обучить все DL-модели и вернуть
словарь предсказаний и метрик."""
    stop =
callbacks.EarlyStopping(monitor="val_loss",
patience=PATIENCE,

restore_best_weights=True)
    results, predictions = {}, {}

    # --- Табличные сети: группированное
разбиение 75/25 по округам ---

```

```

        splitter =
GroupShuffleSplit(n_splits=1,
test_size=config.ML_TEST_SIZE,

random_state=SEED)
        groups = df[config.ID_NAME]
        train_ix, test_ix =
next(splitter.split(df, groups=groups))
        scaler =
StandardScaler().fit(df.iloc[train_ix][feature
s])
        x_tr =
scaler.transform(df.iloc[train_ix][features])
        x_te =
scaler.transform(df.iloc[test_ix][features])
        y_tr =
df.iloc[train_ix][target].to_numpy()
        y_te =
df.iloc[test_ix][target].to_numpy()

        for builder in (build_simple_tf,
build_deep_tf, build_residual_net):
            net = builder(len(features))
            net.fit(x_tr, y_tr,
validation_split=0.15, epochs=EPOCHS,
batch_size=BATCH,
verbose=0, callbacks=[stop])
            pred = net.predict(x_te,
verbose=0).ravel()
            results[net.name] = _metrics(y_te,
pred)
            predictions[net.name] = (test_ix,
pred)
            print(f"    {net.name:12s}
{results[net.name]}")

        # --- Рекуррентные сети: разбиение по
времени ---

```

```

        x_seq, y_seq, yrs = make_sequences(df,
features, target, window=3)
        cut = np.quantile(yrs, 0.75)
# поздние годы - в тест
        tr, te = yrs <= cut, yrs > cut
        flat = x_seq.reshape(-1, x_seq.shape[-
1])
        seq_scaler =
StandardScaler().fit(flat[np.repeat(tr,
x_seq.shape[1])])
        x_seq =
seq_scaler.transform(flat).reshape(x_seq.shape
)

        for flag, label in ((False, "LSTM"),
(True, "BiLSTM")):
            net = build_lstm(x_seq.shape[1],
x_seq.shape[2], bidirectional=flag)
            net.fit(x_seq[tr], y_seq[tr],
validation_split=0.15, epochs=EPOCHS,
                    batch_size=BATCH,
verbose=0, callbacks=[stop])
            pred = net.predict(x_seq[te],
verbose=0).ravel()
            results[label] =
_metrics(y_seq[te], pred)
            print(f"    {label:12s}
{results[label]}")

        table = pd.DataFrame(results).T
        table.index.name = "Model"
        table.to_excel(outdir /
"dl_metrics.xlsx")
        return {"metrics": results,
"predictions": predictions,
                "holdout": (test_ix, y_te)}

```

2.7.11 Модуль *ensembles.py*

```
# -*- coding: utf-8 -*-
"""
    Ансамблирование моделей (глава 6): гибрид
    RF+XGB, объединение ML+DL
    и полный ансамбль. Комбинирование
    выполняется на out-of-fold прогнозах,
    чтобы веса не подгонялись под тестовую
    выборку.
    """

import numpy as np
import pandas as pd

from sklearn.ensemble import
RandomForestRegressor,
GradientBoostingRegressor
    from sklearn.linear_model import RidgeCV
    from sklearn.model_selection import
GroupKFold
    from sklearn.metrics import
mean_absolute_error, mean_squared_error,
r2_score

import config

try:
    xgboost - опционально
    from xgboost import XGBRegressor
    HAS_XGB = True
except ImportError:
    HAS_XGB = False

def _base_models():
    """Базовые модели гибрида; без xgboost
    - GradientBoosting из sklearn."""
```

```

        rf =
RandomForestRegressor(**config.RF_PARAMS)
        if HAS_XGB:
            booster = XGBRegressor(
                n_estimators=600,
learning_rate=0.03, max_depth=6,
                subsample=0.8,
colsample_bytree=0.8,

random_state=config.ML_RANDOM_STATE, n_jobs=-
1,
            )
        else:
            booster =
GradientBoostingRegressor(
                n_estimators=500,
learning_rate=0.05, max_depth=3,

random_state=config.ML_RANDOM_STATE,
            )
        return {"RF": rf, "XGB": booster}

def _metrics(y_true, y_pred) -> dict:
    mse = mean_squared_error(y_true,
y_pred)
    return {
        "R2": round(r2_score(y_true,
y_pred), 3),
        "RMSE": round(float(np.sqrt(mse)),
3),
        "MAE":
round(mean_absolute_error(y_true, y_pred), 3),
    }

def out_of_fold(df, features, target,
n_splits: int = 5):

```

```

        """OOF-прогнозы базовых моделей с
        группировкой по округам."""
        x = df[features].to_numpy()
        y = df[target].to_numpy()
        groups = df[config.ID_NAME].to_numpy()
        folds = GroupKFold(n_splits=n_splits)
        oof = {name: np.zeros(len(df)) for name
in _base_models()}
        for train_ix, val_ix in folds.split(x,
y, groups):
            for name, model in
_base_models().items():
                model.fit(x[train_ix],
y[train_ix])
                oof[name][val_ix] =
model.predict(x[val_ix])
        return oof, y

```

```

def hybrid_rf_xgb(oof: dict, y: np.ndarray)
-> tuple:
    """Гибрид RF+XGB: веса обратно
    пропорциональны MSE каждой модели."""
    inv = {n: 1.0 / mean_squared_error(y,
p) for n, p in oof.items()}
    total = sum(inv.values())
    weights = {n: v / total for n, v in
inv.items()}
    blended = sum(weights[n] * oof[n] for n
in oof)
    return blended, weights

```

```

def stack_ml_dl(oof: dict, dl_preds: dict,
y: np.ndarray,
index: np.ndarray) ->
np.ndarray:

```

"""Стэкинг ML+DL: гребневая регрессия
поверх прогнозов всех моделей.

dl_preds - прогнозы DL-моделей на
удержанной выборке (dl_models.run);
сопоставление выполняется по индексам
наблюдений `index`.

```
"""
    columns = [oof[n][index] for n in
sorted(oof)]
    columns += [p for _, p in
sorted(dl_preds.items())]
    meta_x = np.column_stack(columns)
    meta = RidgeCV(alphas=np.logspace(-3,
2, 25))
    meta.fit(meta_x, y[index])
    return meta.predict(meta_x)
```

```
def run(df, outdir, features, target,
dl_result=None) -> pd.DataFrame:
    """Построить все ансамбли и сохранить
сводную таблицу метрик."""
    oof, y = out_of_fold(df, features,
target)
    rows = {}

    for name, pred in oof.items():
        rows[name] = _metrics(y, pred)

    blended, weights = hybrid_rf_xgb(oof,
y)
    rows["Гибрид RF+XGB"] = _metrics(y,
blended)
    print("  веса гибрида:", {k: round(v,
3) for k, v in weights.items()})

    if dl_result is not None:
```

```

        test_ix, y_te =
dl_result["holdout"]
        dl_preds = {n: p for n, (ix, p) in
dl_result["predictions"].items()}
        stacked = stack_ml_dl(oof,
dl_preds, y, test_ix)
        rows["Ансамбль ML+DL"] =
_metrics(y_te, stacked)

        full_cols = [blended[test_ix]] + [p
for p in dl_preds.values()]
        full =
np.column_stack(full_cols).mean(axis=1)
        rows["Полный ансамбль"] =
_metrics(y_te, full)

        table = pd.DataFrame(rows).T
        table.index.name = "Model"
        table.to_excel(outdir /
"ensemble_metrics.xlsx")
        print(table)
        return table

```

2.7.12 Модуль figures.py

```

# -*- coding: utf-8 -*-
"""
    Построение рисунков главы 3 (3.2.1–3.2.9):
    распределения до/после
    нормализации, корреляционные матрицы,
    диаграммы рассеяния, топ-10 пар,
    распределение и выбросы коэффициентов
    корреляции.

```

Все графики выводятся в шрифте с засечками (гарнитура типа Times New Roman) в разрешении 300 dpi, как в основном тексте Рекомендаций.

```

"""

import numpy as np
import pandas as pd

import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt

import config

plt.rcParams["font.family"] = "serif"
plt.rcParams["axes.unicode_minus"] = False
DPI = 300

STRENGTH_BINS = [0.3, 0.5, 0.7, 0.9]
STRENGTH_LABELS = ["слабая (<0.3)",
"умеренная (0.3-0.5)",
"заметная (0.5-0.7)",
"сильная (0.7-0.9)",
"очень сильная (>0.9)"]

def _save(fig, outdir, name):
    fig.tight_layout()
    fig.savefig(outdir / f"{name}.png",
dpi=DPI)
    plt.close(fig)
    print(" рисунок:", name)

def fig_norm_compare(df, features, outdir,
sample=4):
    """3.2.1 - гистограммы выбранных
признаков до и после нормализации."""
    cols = features[:sample]

```

```

fig, axes = plt.subplots(2, sample,
figsize=(4 * sample, 7))
    for k, col in enumerate(cols):
        raw = df[col].dropna()
        norm = (raw - raw.mean()) /
raw.std(ddof=0)
        axes[0, k].hist(raw, bins=40,
color="skyblue", edgecolor="0.3")
        axes[0, k].set_title(f"{col[:24]}
(до)", fontsize=9)
        axes[1, k].hist(norm, bins=40,
color="salmon", edgecolor="0.3")
        axes[1, k].set_title(f"{col[:24]}
(после)", fontsize=9)
    _save(fig, outdir,
"fig_3_2_1_normalization")

def fig_corr_annotated(df, features,
outdir, threshold=0.7):
    """3.2.2 - матрица переменных,
участвующих в сильных связях."""
    corr = df[features].corr()
    strong = corr.columns[(corr.abs() >
threshold).sum() > 1]
    sub = corr.loc[strong, strong]
    fig, ax = plt.subplots(figsize=(13,
11.5))
    image = ax.imshow(sub, cmap="RdBu_r",
vmin=-1, vmax=1)
    for i in range(len(sub)):
        for j in range(len(sub)):
            val = sub.iloc[i, j]
            ax.text(j, i, f"{val:.2f}",
ha="center", va="center",
                    fontsize=5,
color="white" if abs(val) > 0.55 else "black")
    ax.set_xticks(range(len(sub)))

```

```

        ax.set_yticks(range(len(sub)))
        ax.set_xticklabels(sub.columns,
rotation=90, fontsize=7)
        ax.set_yticklabels(sub.columns,
fontsize=7)
        fig.colorbar(image, ax=ax,
fraction=0.046)
        _save(fig, outdir,
"fig_3_2_2_corr_strong")

def fig_corr_classes(df, features, outdir):
    """3.2.3 - нижний треугольник с
    классами силы связи и значениями r."""
    corr = df[features].corr().to_numpy()
    classes = np.digitize(np.abs(corr),
STRENGTH_BINS).astype(float)
    masked = np.ma.array(classes,
mask=np.triu(np.ones_like(corr), 1))
    fig, ax = plt.subplots(figsize=(15,
13.5))
    image = ax.imshow(masked,
cmap="RdYlBu_r", vmin=0, vmax=4)
    n = len(features)
    for i in range(n):
        for j in range(i + 1):
            ax.text(j, i, f"{corr[i,
j]:.2f}".replace("0.", "."),
                    ha="center",
va="center", fontsize=3.5,
                    color="white" if
classes[i, j] in (0, 4) else "black")
    ax.set_xticks(range(n));
    ax.set_yticks(range(n))
    ax.set_xticklabels(features,
rotation=90, fontsize=5)
    ax.set_yticklabels(features,
fontsize=5)

```

```

        bar = fig.colorbar(image, ax=ax,
ticks=range(5), fraction=0.035)
        bar.ax.set_yticklabels(STRENGTH_LABELS,
fontsize=8)
        _save(fig, outdir,
"fig_3_2_3_corr_classes")

```

```

def top_pairs(df, features, k=10) ->
pd.DataFrame:
    """Топ-к пар по модулю корреляции (без
дублей и диагонали)."""
    corr = df[features].corr()
    upper =
corr.where(np.triu(np.ones(corr.shape),
1).astype(bool))
    pairs =
upper.stack().rename("r").reset_index()
    pairs["abs_r"] = pairs["r"].abs()
    return pairs.nlargest(k,
"abs_r").reset_index(drop=True)

```

```

def fig_scatter_pairs(df, outdir, pairs,
n_show=6):
    """3.2.5 - диаграммы рассеяния для пар
с наибольшей |r|."""
    fig, axes = plt.subplots(2, 3,
figsize=(15, 9))
    for ax, (_, row) in zip(axes.ravel(),
pairs.head(n_show).iterrows()):
        x, y = df[row["level_0"]],
df[row["level_1"]]
        ax.plot(x, y, ".", ms=3)

ax.set_xlabel(str(row["level_0"])[32],
fontsize=8)

```

```

ax.set_ylabel(str(row["level_1"])[32],
fontsize=8)
        ax.set_title(f"r = {row['r']:.3f}",
fontsize=10)
        ax.grid(color="0.9")
        _save(fig, outdir, "fig_3_2_5_scatter")

def fig_top10(df, outdir, pairs):
    """3.2.6 - горизонтальная диаграмма
    топ-10 пар."""
    fig, ax = plt.subplots(figsize=(12.9,
6.6))
    labels = [f"{a[:22]}... ↔ {b[:22]}..."
               for a, b in
zip(pairs["level_0"], pairs["level_1"])]
    colors = ["#3742fa" if r > 0 else
"#ff4757" for r in pairs["r"]]
    ax.barh(range(len(pairs)), pairs["r"],
color=colors)
    ax.set_yticks(range(len(pairs)))
    ax.set_yticklabels(labels, fontsize=8)
    ax.axvline(0, color="0.2", lw=0.8)
    ax.set_xlabel("Коэффициент корреляции
Пирсона (r)")
    _save(fig, outdir, "fig_3_2_6_top10")

def fig_corr_distribution(df, features,
outdir):
    """3.2.7 и 3.2.8 - распределение всех
    парных корреляций."""
    corr = df[features].corr().to_numpy()
    values =
corr[np.triu_indices(len(features), 1)]
    values = values[~np.isnan(values)]

```

```

        fig, axes = plt.subplots(2, 2,
figsize=(12.9, 8.4))
        box_ax, violin_ax, qq_ax, rank_ax =
axes.ravel()
        box_ax.boxplot(values, vert=False)
        box_ax.set_title("Box Plot корреляций")
        violin_ax.violinplot(values)
        violin_ax.set_title("Violin Plot
корреляций")
        sorted_vals = np.sort(values)
        theor =
np.sort(np.random.default_rng(0).normal(
        sorted_vals.mean(),
sorted_vals.std(), len(sorted_vals)))
        qq_ax.plot(theor, sorted_vals, ".",
ms=2)
        qq_ax.plot(theor, theor, "r-", lw=1)
        qq_ax.set_title("Q-Q Plot корреляций")
        rank_ax.plot(np.arange(1, len(values) +
1), sorted_vals, ".", ms=3)
        rank_ax.set_title("Корреляции vs
Ранги")
        _save(fig, outdir,
"fig_3_2_7_outliers")

        fig, ax = plt.subplots(figsize=(9.6,
6.6))
        ax.hist(values, bins=40,
color="skyblue", edgecolor="0.3")
        ax.set_xlabel("Коэффициент корреляции")
        ax.set_ylabel("Частота")
        _save(fig, outdir,
"fig_3_2_8_histogram")

```

```

def run(df, outdir, features):
    """Построить полный комплект рисунков
главы 3."""

```

```

        fig_norm_compare(df, features, outdir)
        fig_corr_annotated(df, features,
outdir)
        fig_corr_classes(df, features, outdir)
        pairs = top_pairs(df, features, k=10)
        fig_scatter_pairs(df, outdir, pairs)
        fig_top10(df, outdir, pairs)
        fig_corr_distribution(df, features,
outdir)

```

2.7.13 Модуль *pipeline_full.py*

```

# -*- coding: utf-8 -*-
"""
Расширенный сквозной конвейер: базовые
этапы run_all.py плюс глубокое
обучение (глава 5), ансамбли (глава 6) и
комплект рисунков главы 3.

```

```

Запуск: python pipeline_full.py
Дополнительные выгрузки в outputs/:
    dl_metrics.xlsx          - качество DL-
моделей (R2, RMSE, MAE, MSE)
    ensemble_metrics.xlsx    - качество
гибридов и ансамблей
    fig_3_2_*.png           - рисунки
корреляционного блока
"""

```

```

import numpy as np
import pandas as pd

import config
import apk_io
import correlation
import integral_index
import dea_analysis
import ml_drivers

```

```

import figures

try:
    import dl_models
    HAS_TF = True
except ImportError:                                #
TensorFlow не установлен
    HAS_TF = False

try:
    import data_gsheets
    LOADER = data_gsheets.load_or_cache
except ImportError:
    LOADER = apk_io.load_dataset

def numeric_features(df: pd.DataFrame) ->
list:
    """Все числовые показатели, кроме
служебных полей и целевой."""
    skip = {config.ID_NAME, config.ID_YEAR,
config.ML_TARGET}
    return [c for c in df.columns
            if c not in skip and
pd.api.types.is_numeric_dtype(df[c])]

def main():
    outdir = apk_io.ensure_outdir()
    print("=" * 70)
    print("ИНТЕГРАЛЬНАЯ ОЦЕНКА РАЗВИТИЯ АПК
- РАСШИРЕННЫЙ КОНВЕЙЕР")
    print("=" * 70)

    df = LOADER()
    features = numeric_features(df)
    print(f"Панель: {df.shape[0]}
наблюдений, {len(features)} признаков")

```

```

        print("\n--- 1. Корреляционный анализ и
рисунки ---")
        correlation.run(df, outdir)
        figures.run(df, outdir, features)

    print("\n--- 2. Интегральный индекс ---
")
    df = integral_index.run(df, outdir)

    print("\n--- 3. DEA (эффективность АПК)
---")
    dea_analysis.run(df, outdir)

    print("\n--- 4. Драйверы (Random Forest
+ SHAP) ---")
    ml_drivers.run(df, outdir)

    target = config.ML_TARGET
    dl_result = None
    if HAS_TF:
        print("\n--- 5. Глубокое обучение
(SimpleTF/DeepTF/ResNet/LSTM) ---")
        dl_result = dl_models.run(df,
outdir, features, target)
    else:
        print("\nTensorFlow не найден - DL-
блок пропущен.")

    print("\n--- 6. Ансамбли (RF+XGB,
ML+DL, полный) ---")
    import ensembles
    ensembles.run(df, outdir, features,
target, dl_result)

    print("\nГотово. Результаты - в
каталоге:", outdir)

```

```
if __name__ == "__main__":
    main()
```

2.7.14 Модуль *run_all.py*

```
# -*- coding: utf-8 -*-
"""
```

Полный пайплайн интегральной оценки развития АПК по панельным данным «территория-год» (датасет подставляется пользователем).

Последовательность:

1. загрузка входного датасета;
2. корреляционный анализ (Пирсон/Спирмен/Кендалл);
3. интегральный индекс (нормировка -> энтропийные веса -> агрегирование);
4. DEA-эффективность (CCR, супер-эффективность, Мальмквист);
5. драйверы интегральной оценки (Random Forest + SHAP).

Запуск: `python run_all.py`
 Результаты складываются в каталог `outputs/`.

```
"""
import config
import apk_io
import correlation
import integral_index
import dea_analysis
import ml_drivers
```

```
def main():
    outdir = apk_io.ensure_outdir()
    print("=" * 70)
```

```

        print("ИНТЕГРАЛЬНАЯ ОЦЕНКА РАЗВИТИЯ АПК
- ПОЛНЫЙ МАССИВ")
        print("=" * 70)

        df = apk_io.load_dataset()

        print("\n--- 1. Корреляционный анализ -
--")
        correlation.run(df, outdir)

        print("\n--- 2. Интегральный индекс ---
")
        df = integral_index.run(df, outdir)

        print("\n--- 3. DEA (эффективность АПК)
---")
        dea_analysis.run(df, outdir)

        print("\n--- 4. Драйверы (Random Forest
+ SHAP) ---")
        ml_drivers.run(df, outdir)

        print("\nГотово. Результаты - в
каталоге:", outdir)

if __name__ == "__main__":
    main()

```

2.7.15 Файл requirements.txt

```

# Базовые (нужны всегда)
pandas>=1.3
numpy>=1.21
openpyxl>=3.0

```

```

# Машинное обучение (раздел драйверов)

```

```
scikit-learn>=1.0

# Интерпретация моделей (опционально; без
него SHAP-блок пропускается)
shap>=0.41

# Графики (опционально)
matplotlib>=3.4

# Глубокое обучение и ансамбли
(опционально; главы 5-6)
tensorflow>=2.10
xgboost>=1.6

# Примечание: DEA реализована на чистом
NumPy (dea_lp.py),
# поэтому scipy НЕ требуется.
```

Учебное издание

О.Ә. Әліпбеки, П.П. Гроссул,
Ч.А. Аліпбекова, Д.К. Рахимов, М.М. Алиев,
Г. Мусайф, Қ.К. Аугамбаев

**РЕКОМЕНДАЦИИ ПО ИНТЕГРАЛЬНОЙ ОЦЕНКЕ
РАЗВИТИЯ АПК НА ОСНОВЕ ПРОСТРАНСТВЕННО-
ВРЕМЕННЫХ, СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ И
ЭКОЛОГИЧЕСКИХ ДАННЫХ**

Подписано в печать 2.09.2026.
Формат 60x84/16.
Печать цифровая. Объем 12,25 п.л.
Тираж 50 экз.

Отпечатано в типографии «ESKoPrint».

ISBN 978-601-04-7643-1



9 786010 476431