**ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»**
**ПО ПРОФИЛЮ «ИНЖЕНЕРНОЕ ДЕЛО»**

*регистрационный номер*

**Секция:** Информационные технологии

# Разработка геоинформационной базы данных с использованием PostgreSQL для хранения данных спутникового мониторинга сельскохозяйственных полей Белогорского района Амурской области

**Автор:** Заварыкина Татьяна Денисовна,
г. Хабаровск, МБОУ СОШ №68, 11Б
**Научный руководитель:** Илларионова Любовь Викторовна,
старший научный сотрудник ВЦ ДВО РАН,
кандидат физико-математических наук

**Хабаровск – 2024**

**Аннотация**

Разработана ГИС, позволяющая обрабатывать и хранить данные дистанционного зондирования Земли, полученные с помощью спутника Sentenel-2. Система состоит из базы данных, которая использует PostgreSQL, блока обработки данных, блока визуализации данных и блока классификации сельскохозяйственных культур. Система может быть использована специалистами в области сельского хозяйства для мониторинга и анализа состояния сельскохозяйственных полей в Белогорском районе Амурской области.

Объектом исследования являются сельскохозяйственные поля Белогорского района Амурской области РФ.

***Цель работы***: разработка ГИС для обработки и хранения данных дистанционного зондирования и результатов классификации сельскохозяйственных культур Белогорского района Амурской области.

Во время разработки были использованы методы обработки и анализа геопространственных данных, машинного обучения (модели наивного байесовского классификатора и использование градиентного бустинга), а также методы построения геоинформационных карт и визуализации данных.

Основные результаты:

1. Разработана архитектура информационной системы, включая базу данных для хранения геопространственных данных о сельскохозяйственных угодьях.
2. Реализованы инструменты для сбора и систематизации данных, а также для визуализации информации о сельскохозяйственных полях.
3. Построена модель машинного обучения для классификации культур с использованием градиентного бустинга.
4. Проведена оценка производительности модели.

Содержание

## Введение

Сегодня интеграция цифровых технологий выступает в качестве ключевой силы, изменяющей традиционные методы ведения сельского хозяйства. Цифровизация в сельском хозяйстве охватывает целый ряд технологий, среди которых агрегирование и анализ данных, помогающие принимать решения по оптимизации урожайности, а также использование дистанционного зондирования и спутниковый мониторинг, который позволяет собирать данные о состоянии урожая, индексах растительности, составе почвы и др. Технологии дистанционного зондирования помогают в создании подробных карт, определении границ полей, мониторинге изменений в землепользовании и выявлении областей, требующих конкретных вмешательств. Эффективное управление данными спутникового мониторинга имеет важное значение для оптимизации методов ведения современного сельского хозяйства [1]. Существует потребность в надёжной системе баз данных (БД), специально разработанной для хранения и управление большими объемами геопространственной информации, относящейся к сельскохозяйственным полям.

Цель работы: разработка ГИС для обработки и хранения данных дистанционного зондирования и результатов классификации сельскохозяйственных культур Белогорского района Амурской области.

Задачи работы:

1. Разработка архитектуры информационной системы, подбор необходимого программного обеспечения, и создание логической и структурной схем базы данных.
2. Сбор, систематизация и внесение данных о сельскохозяйственных угодьях Белогорского района в базу данных; сбор спектральных характеристик спутника Sentinel-2; расчет и аппроксимация вегетационных индексов для каждого пикселя исследуемых полей.
3. Разработка инструментов для фильтрации и визуализации данных.

4. Построение модели машинного обучения с целью классификации сельскохозяйственных культур, а также оценка точности полученной модели.
5. Занесение результатов классификации в базе данных и размещение её в сети Интернет для обеспечения доступа стейкхолдерам и заинтересованным лицам.

Заранее произведено структурное разделение создаваемой геоинформационной системы на четыре модуля. Схема представлен на рисунке 1.


*Рис. 1.* Структура ГИС

# 1 ОСНОВНАЯ ЧАСТЬ

## 1.1 База данных

### 1.1.1 Разработка структуры и процесс реализации базы данных

Основополагающей разработкой для любой геоинформационной системы (ГИС) является база данных (БД). В работе рассмотрен процесс разработки и настойки БД для обеспечения эффективности работы ГИС.

Использование БД для хранения геоданных имеет ряд преимуществ:

- БД позволяет эффективно хранить и обрабатывать большие объемы геоданных, что важно при работе с большими наборами данных,
- удобные инструменты для управления данными, включая возможность выполнять запросы SQL для извлечения и обработки данных, а также для импорта и экспорта данных,
- простая интеграция геоданных с другими инструментами и системами,
- снабжение доступа к данным на разных устройствах и для разных пользователей через сеть.

Для создания БД было принято решение использовать мощную систему управления базами данных PostgreSQL 15. PostgreSQL позволяет поддерживать большое количество типов данных, имеет возможность подключать к работе БД язык программирования Python и имеет множество встроенных функций, таких как представления, триггеры и хранимые процедуры. Данная система была выбрана из-за гибкости, производительности и поддержки различных расширений, таких как PostGIS. PostGIS – это расширение PostgreSQL, которое добавляет поддержку географических объектов. Это позволяет хранить геоинформационные данные в БД и выполнять сложные пространственные запросы. Использование PostGIS при реализации ГИС дало возможность использования методов географического кодирования и декодирования, что облегчило создание БД. Для предварительной обработки и создания данных было использовано такое программное обеспечение, как QGIS и DBeaver

Ultimate, для работы с запросами к БД была использована программа Visual Studio Code и язык программирования Python.

Структурно вся БД делится на несколько пространств: SupplementaryData и схемы вида “YEAR”. Модуль SupplementaryData содержит такие неизменяемые данные, как информация о землях сельскохозяйственного назначения, включая списки владельцев полей, виды сельскохозяйственных культур и сорта различных культур, произрастающих в Белогорском районе. Именно эти данные являются едиными для всей БД. Модуль вида “YEAR” представлен в БД несколько раз, поскольку каждый такой модуль служит блоком актуальных данных для конкретного года. Этот подход позволяет эффективно управлять данными, не затрагивая другие схемы, и обеспечивает четкое разделение данных по годам. Пространство вида “YEAR” содержит основную информацию о полях: данные о ожидаемой и действительной произрастающей культуре, об владельце поля, а также такие геоданные, как форма поля в виде мультиполигона. Помимо этого, в этом пространстве хранится информация о вегетационных индексах. Эта информация рассчитывается для каждой точки полей Белогорского района для некоторого промежутка недель в году. Все точки имеют свои геоданные и множество показателей вегетационных индексов, а также связаны с данными поля, на котором располагается точка. На данный момент в БД хранится информация только об индексах вегетации NDVI, но благодаря скриптовой реализации существует возможность быстро интегрировать и другие индексы вегетации.

NDVI, или Normalized Difference Vegetation Index, — один из самых известных вегетационных индексов. Он определяется через коэффициенты спектральной яркости для красной и инфракрасной областей спектра по следующей формуле:

$$NDVI = \frac{NIR-RED}{NIR+RED},$$

где RED – поглощение и отражение красных лучей, NIR – поглощение и отражение инфракрасных лучей. Выбор именно этих участков определяется, в первую очередь, оптическими свойствами хлорофилла. В диапазоне длин волн 600–700 нм, то есть в зоне видимого красного, он имеет выраженный максимум поглощения. По этой же причине все содержащие хлорофилл части растений имеют зеленый цвет. В области ближнего ИК-излучения растительность обычно обладает высоким коэффициентом отражения. В совокупности эти два факта означают, что можно судить о концентрации хлорофилла в каком-либо объекте, сравнивая его отражательные характеристики в двух указанных диапазонах. Формула в виде нормализованной разности удобна для расчета компьютерными средствами, поскольку множество значений функции имеет конечные значения.

Данные для расчета NDVI можно получить из нескольких источников: спутниковой съемки, наземных датчиков, а также фотографирования с БЛА. В нашей работе значение индекса получено со снимков спутника Sentenel - 2. Подробная схема БД представлена на рисунке 2.

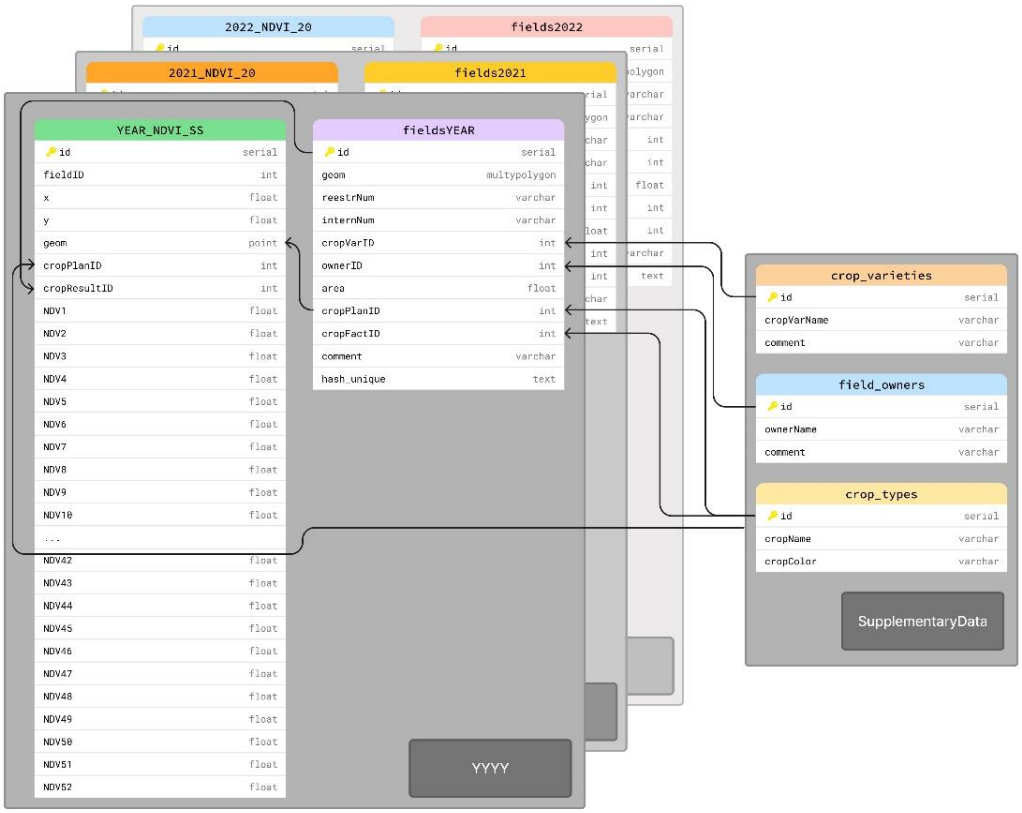


*Рис. 2.* Схема геоинформационной базы данных

### 1.1.2 Функции БД, фильтрация и визуализация геоданных

Для облегчения занесения данных в таблицы БД были реализованы функции, реализованные на основе расширения PostGIS и вызываемые по команде триггера. При занесении данных в таблицу вида “fieldsYEAR”, которая ориентирована на сохранение данных о полях, вызывается функция расчета площади поля по данным о его геометрии. При занесении данных в таблицу вида “YEAR_NDVI_SS” также вызываются две функции: createPointFromXY() и transferPlanCrop(). Первая функция создает объект геометрии Point, группа которых в подлеющем может использоваться для отрисовки данных в картографическом программном обеспечении, и заносит это значение в определенную ячейку. Вторая функция реализует поиск и вставку планируемой культуры произрастающих растений из таблицы вида “fieldsYEAR” в данную таблицу.

Помимо триггерных функций, была реализована функция calculatePlotAvg() для таблицы вида “YEAR_NDVI_SS”. Функция, основываясь на введенных данных, вычисляет среднее значение показателя NDVI для всего промежутка времени. На основе этих данных строится график, который показывает изменение среднего значения NDVI во времени для выбранных полей и/или культур растений. Данная функция генерируется при создании новой таблицы вида “YEAR_NDVI_SS”. Функция принимает 3 аргумента: название культуры, id поля и дополнительное условие отбора. Далее функция вычисляет средние значения NDVI для всех точек, которые соответствуют заданным параметрам, и строит график зависимости среднего показателя от времени. Это может быть полезно для анализа изменений в состоянии различных культур, поиска выбросов или для сравнения состояния растений на различных полях или с различными культурами растений. Примеры использования функции представлены на рисунке 3.

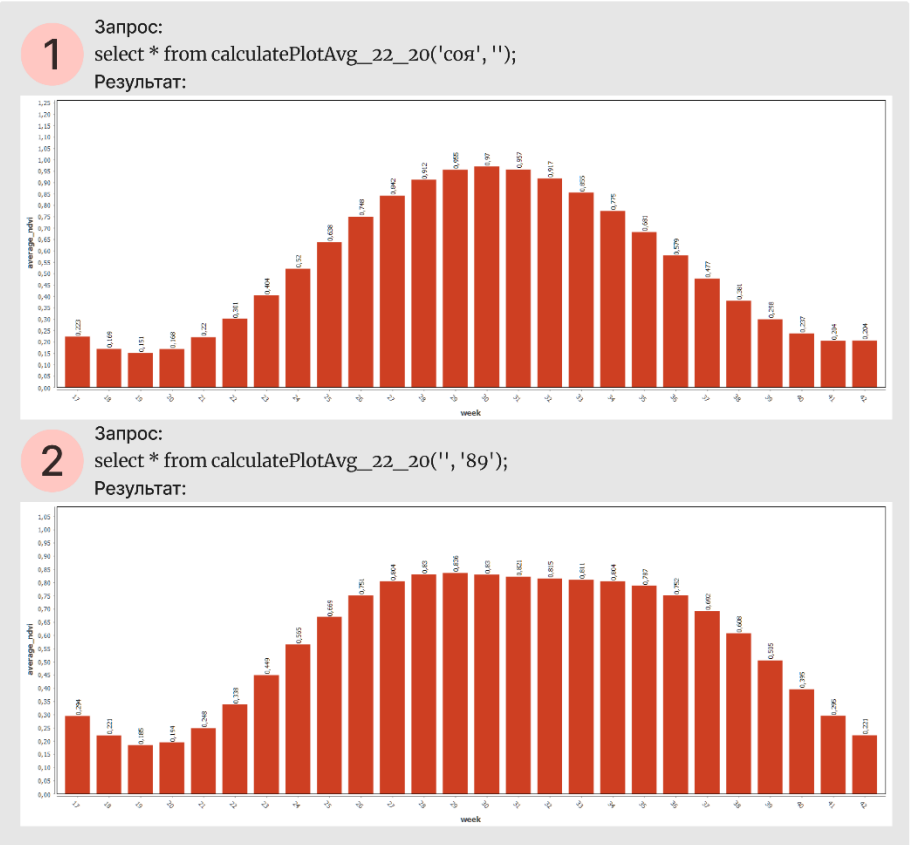


*Рис. 3.* Пример применения функции 1) с отбором по культуре, 2) с отбором по id поля

Геоданные, представленные в БД, могут быть визуализированы в свободно распространяемой системе QGIS, что позволяет проводить более глубокий анализ данных. С помощью QGIS можно создавать карты, которые могут отражать различные аспекты географии Белогорского района, такие как границы, расположение населенных пунктов, типы почвы, а также использовать данные созданной ГИС, возможна фильтрация и классификация полей и точек по показателям культур растений и т. д. Пример использования QGIS для визуализации представлен на рисунке 4.

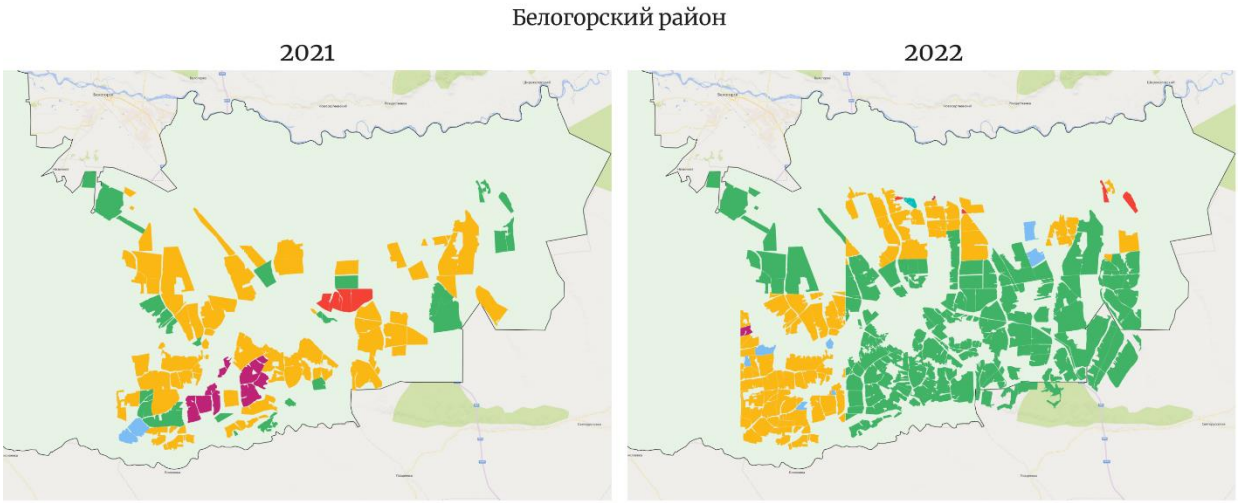


*Рис. 4.* Визуализация при помощи QGIS части полей Белогорского района за 2021 и 2022 год

Создание базы данных позволить хранить информацию о данных дистанционного зондирования земель региона и будет способствовать решению задач цифрового земледелия для Белогорского района Амурской области.

### 1.2 Классификация сельскохозяйственных культур

Современные технологии в области земледелия играют ключевую роль в оптимизации управления сельскими хозяйствами и повышении урожайности. В этом контексте, разработка и внедрение современных методов анализа данных становится неотъемлемой частью сельскохозяйственной практики. В данной части проекта предлагается рассмотреть проблему классификации сельскохозяйственных культур на основе временных рядов индекса нормализованной разности вегетации, полученных с помощью спутниковых наблюдений. Процесс классификации сельскохозяйственных культур является важным инструментом для анализа состояния посевов, определения роста и развития растений, а также для принятия решений в области управления земельными ресурсами.

В рамках исследования, посвященного классификации сельскохозяйственных культур, были изучены два алгоритма машинного обучения, представленные на рисунке 5.


*Рис. 5.* Блок-схемы предлагаемых методов классификации

Методы классификации:

- градиентный бустинг с категориальной обработкой,
- наивный байесовский метод Гаусса.

### 1.2.1 Предварительная обработка данных

Для анализа состояния растительности на сельскохозяйственных участках были использованы данные, полученные с помощью спутника Sentinel-2. Этот спутник обеспечивает широкий охват территории и высокое пространственное разрешение, что позволяет получать детальную информацию о состоянии посевов. Сначала были собраны группы снимков со спутника Sentinel-2 для периода с конца апреля по конец октября 2021 и 2022 годов, чтобы охватить весь цикл жизни растений, начиная от момента посева и заканчивая уборкой урожая. Далее, после проведения маскирования облачности, теней от них и снега, а также исключения или коррекции данных с облачностью для обеспечения точности и надежности вычислений, были извлечены спектральные полосы красного (RED) и инфракрасного (NIR) диапазонов, которые содержат информацию о различных длинах волн света. Для каждого пикселя изучаемых полей были сформированы временные ряды вегетационного индекса NDVI.

Для сглаживания временных рядов NDVI были применены методы аппроксимации данных, что позволило получить более стабильные и надежные результаты анализа. Полученные данные в таком формате были внесены в базу

данных для дальнейшего использования в анализе и классификации сельскохозяйственных культур [2].

Для реализации классификации сельскохозяйственных культур на основе временных рядов NDVI было принято использовать методы машинного обучения. Эти методы позволят определить, к какой сельскохозяйственной культуре относится определенная точка данных на основе ее временного ряда NDVI. Для разработки и обучения моделей классификации был выбран язык программирования Python, который предоставляет широкий спектр инструментов и библиотек для работы с данными и машинным обучением. Некоторые из ключевых библиотек, использованных в этом проекте, включают в себя: NumPy для работы с массивами данных, Pandas для удобной работы с табличными данными, Scikit-learn для реализации различных алгоритмов машинного обучения, UMAP для снижения размерности данных, CatBoost для градиентного бустинга и др. Эти инструменты позволяют эффективно обрабатывать данные, создавать и обучать модели классификации, а также оценивать их производительность.


*Рис. 6.* Используемые модули Python для реализации классификации

### 1.2.2 Обработка данных для обучения и построение моделей

Для подготовки данных к обучению моделей классификации применяется предварительная обработка индекса нормализованной разности вегетации. Экстремальные значения NDVI фильтруются и заменяются на пустые значения. Это необходимо, так как индекс NDVI измеряется в диапазоне от -1 до 1, и экстремальные значения могут исказить результаты обучения моделей.

Для заполнения пустых значений NDVI в датасете был выбран метод ближайших соседей (K-Nearest Neighbor). Этот метод позволяет использовать информацию о близких наблюдениях для заполнения отсутствующих данных, что способствует сохранению общих трендов и структуры данных. В нашем случае данный алгоритм был реализован с помощью библиотеки scikit-learn языка программирования Python с параметром k=5. Параметр k=5 означает, что мы рассматриваем пять ближайших соседей для каждого пропущенного значения. Значения этих соседей взвешиваются по расстоянию до пропущенного значения, где ближайшие соседи имеют больший вес. Затем используется среднее значение взвешенных соседей для заполнения пропущенного значения NDVI.

Для снижения размерности временных рядов NDVI и улучшения вычислительной эффективности анализа выбран алгоритм UMAP (Uniform Manifold Approximation and Projection). UMAP использует метод стохастического градиентного спуска для построения низкоразмерного представления данных, сохраняя при этом глобальную структуру и схожесть точек в исходном пространстве. Алгоритм был реализован при помощи библиотеки umap-learn на языке Python, параметры алгоритма выбраны с учетом оптимального баланса между точностью и вычислительной сложностью: n_neighbors=250, min_dist=0.15, n_components=2, metric='euclidean'. В результате применения UMAP размерность временных рядов была снижена с 27 до 2, что значительно упрощает последующее обучение моделей машинного обучения.

Данные для обучения моделей представляют собой временные ряды NDVI, аналогичные тем, которые были собраны для Белогорского района Амурской области. Однако, в этом случае данные собраны для соседнего района, где уже известна достоверная информация о том, какие сельскохозяйственные культуры выращиваются на каждом конкретном поле. Для обучения моделей был применен метод кросс-валидации, который предусматривает разделение данных на обучающий и тестовый наборы. В данном случае, 80% от общего числа пикселей (44684 записей) были выделены в обучающую выборку, в то время как оставшиеся 20% (11170 записей) использовались для тестирования моделей. Этот подход позволяет оценивать производительность моделей на различных подмножествах данных, что улучшает обобщение и устойчивость результатов. Кросс-валидация предотвращает переобучение модели на конкретных данных, делая оценку её эффективности более надежной и репрезентативной для общей совокупности данных.

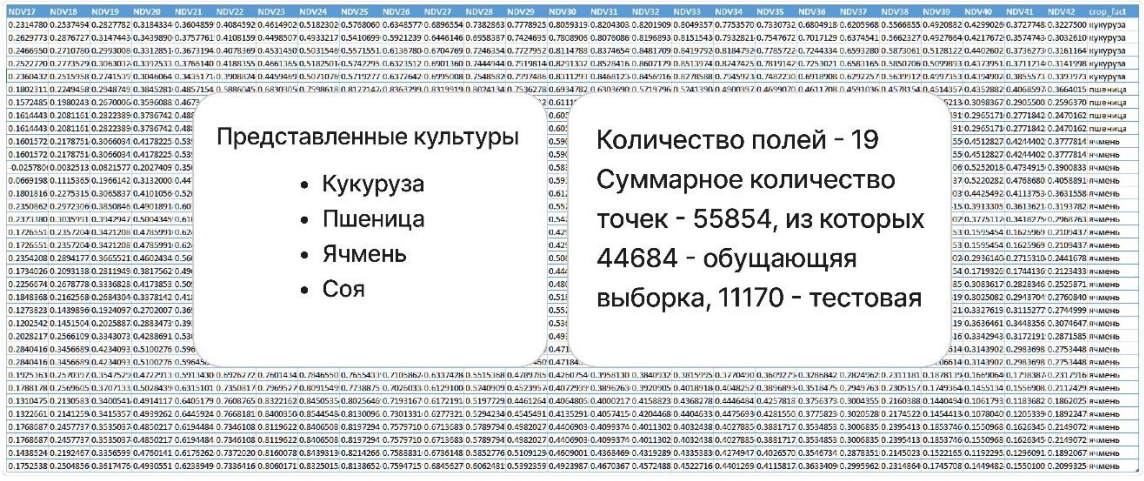


*Рис. 7.* Обзор данных для обучения моделей классификации сельскохозяйственных культур

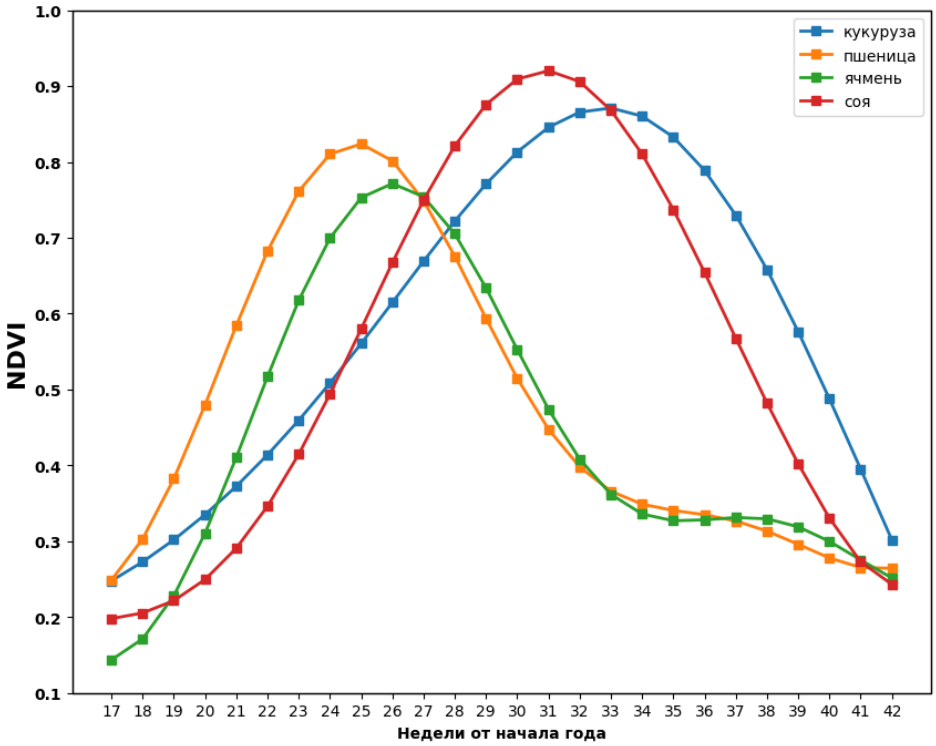


*Рис. 8.* График исходных временных рядов NDVI для с/х культур на протяжении недель от начала года

### 1.2.3 Обучение Наивного байесовского классификатора

Наивный байесовский классификатор — это вероятностный метод классификации, основанный на применении теоремы Байеса с наивным предположением о независимости между признаками. В библиотеке Scikit-learn реализовано несколько вариантов наивных байесовских классификаторов, включая Gaussian Naive Bayes (GaussianNB), который подходит для непрерывных признаков, предполагающих нормальное распределение.

Для определения и оптимизации гиперпараметров модели GaussianNB, таких как var_smoothing (параметр сглаживания), было принято решение использовать библиотеку Optuna в Python. Optuna — это библиотека для оптимизации гиперпараметров и автоматизации процесса настройки модели. Она позволяет проводить поиск оптимальных значений гиперпараметров в заданном пространстве поиска. Optuna итеративно пробует различные значения гиперпараметров GaussianNB, оценивает их производительность с помощью кросс-валидации, и находит оптимальные значения. При помощи этой

библиотеки было определено лучшее значение гиперпараметра var_smoothing=1.030250175573304e-12.

Экземпляр классификатора GaussianNB инициализируется с оптимальным значением гиперпараметра var_smoothing, которое было найдено ранее.

Далее, классификатор обучается на обучающем наборе данных (X_train, y_train), где X_train представляет из себя матрицу признаков (временные ряды NDVI), а y_train - вектор целевых меток классов сельскохозяйственных культур. После обучения классификатора получаем предсказания на тестовом наборе данных (X_test) при помощи метода predict. Для оценки качества классификации используется метрика точности (accuracy). Метрика сравнивает фактические значения целевых меток с предсказанными значениями и вычисляет долю правильно классифицированных образцов. Кроме того, выводится отчет о классификации, который предоставляет более подробную информацию о производительности классификатора, включая значения метрик precision, recall и f1-score для каждого класса, а также средневзвешенные значения этих метрик по всем классам. После обучения и оценки модель наивного байесовского классификатора сохраняется в виде файла при помощи модуля pickle в Python. Это позволяется загрузить модель позже из этого файла и использовать ее для предсказаний без необходимости повторного обучения.

```python
# Определение гиперпараметров для оптимизации
def objective(trial):
    var_smoothing = trial.suggest_loguniform('var_smoothing', 1e-12, 1e-5)
    # Инициализация и обучение наивного байесовского классификатора с определенными гиперпараметрами
    naive_bayes_classifier = GaussianNB(var_smoothing=var_smoothing)
    # Оценка точности с использованием кросс-валидации
    accuracy = cross_val_score(naive_bayes_classifier, X_train, y_train, cv=5, scoring='accuracy').mean()
    return accuracy
# Запуск оптимизации
study = optuna.create_study(direction='maximize')
study.optimize(objective, n_trials=100)
# Получение лучших гиперпараметров
best_params = study.best_params
# Инициализация и обучение наивного байесовского классификатора с лучшими гиперпараметрами
best_naive_bayes_classifier = GaussianNB(var_smoothing=best_params['var_smoothing'])
best_naive_bayes_classifier.fit(X_train, y_train)

# Получение предсказаний на тестовом наборе
nb_predictions = best_naive_bayes_classifier.predict(X_test)

# Оценка точности
accuracy = accuracy_score(y_test, nb_predictions)
print(f'Best hyperparameters: {best_params}')
print(f'Accuracy: {accuracy}')
print(classification_report(y_test, nb_predictions))
```

*Рис. 9.* Реализация наивного байесовского классификатора

### 1.2.4 Обучение модели Градиентного бустинга

Градиентный бустинг – это метод машинного обучения, который использует ансамбль деревьев решении для решения задач классификации и регрессии. Данный метод работает путем последовательного добавления деревьев к модели, где каждое новое дерево исправляет ошибки предыдущей модели. В конечном результате получается сильная модель, способная делать достаточно точные прогнозы.

Библиотека CatBoost в Python представляет собой эффективный инструмент для реализации градиентного бустинга, специально разработанный для работы с категориальными признаками. Для исследования данного метода классификации была выбрана модель CatBoostClassifier со следующими параметрами:

- iterations=100: Количество деревьев, которые будут построены в модели.
- depth=6: Максимальная глубина дерева. Каждое дерево имеет не более 6 уровней.
- learning_rate=0.1: Скорость обучения, которая определяет величину шага, с которым обновляются веса модели на каждой итерации. Низкая скорость

обучения хоть и требует большего количества итераций для сходимости, но может помочь избежать переобучения модели.

- loss_function='MultiClass': Функция потерь, используемая для оптимизации модели. В данном случае была выбрана функция потерь для многоклассовой классификации.
- random_seed=42: Зерно случайности для воспроизводимости результатов.

Далее, модель CatBoostClassifier обучается на обучающем наборе данных X_train с соответствующими целевыми метками y_train. Затем получаются предсказания y_pred на тестовом наборе данных X_test, и выводится отчет о классификации с помощью функции classification_report из модуля Scikit-learn. Обученная модель сохраняется в файле catboostclassifier.pkl с помощью модуля pickle. Сохранение модели экономит время и ресурсы, модель легко подгрузить в других приложениях и использовать на новых данных.

### 1.2.5 Сравнительная оценка методов

Обе модели, классификатор градиентного бустинга и наивный байесовский классификатор, были подвергнуты тестированию на отдельной тестовой выборке, которая не использовалась в процессе обучения моделей. Для оценки производительности каждой модели были использованы следующие метрики:

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

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

$$f1 = \frac{TP}{TP+\frac{FP+FN}{2}},$$

где TP (True Positives) представляет собой количество объектов, которые модель правильно классифицировала как положительные; FP (False Positives) — это количество объектов, которые модель неправильно классифицировала как положительные, когда на самом деле они отрицательные; FN (False Negatives) — это количество объектов, которые модель неправильно классифицировала как отрицательные, когда на самом деле они положительные.

После обучения и тестирования моделей был составлен отчет о точности классификации, который включает значения этих метрик для каждого изучаемого сельскохозяйственного класса, а также общую точность для всего набора данных, который изображен на рисунке 10.

Наивный байесовский классификатор (GaussianNB)

| | Precision | Recall | f1-score |
|---|---|---|---|
| Кукуруза | 0.65 | 0.76 | 0.70 |
| Пшеница | 0.90 | 0.99 | 0.94 |
| Соя | 0.97 | 0.95 | 0.96 |
| Ячмень | 0.93 | 0.47 | 0.62 |
| | | | |
| Общая точность | | | 0.92 |

Градиентный бустинг (CatBoostClassifier)

| | Precision | Recall | f1-score |
|---|---|---|---|
| Кукуруза | 0.92 | 0.80 | 0.86 |
| Пшеница | 0.96 | 0.97 | 0.97 |
| Соя | 0.98 | 0.99 | 0.98 |
| Ячмень | 0.88 | 0.81 | 0.85 |
| | | | |
| Общая точность | | | 0.96 |

*Рис. 10.* Сравнительный отчет с метриками для двух моделей машинного обучения

Результаты тестирования модели, основанной на алгоритме градиентного бустинга, демонстрируют высокую точность классификации на уровне 96%, а также сбалансированные значения метрик precision, recall и F1-score для каждого класса. Эти результаты свидетельствуют о высокой эффективности модели в решении задачи классификации сельскохозяйственных культур.

Модель GaussianNB показала более низкую общую точность классификации на уровне 92% и выявила различия в метриках между разными классами. Особенно заметны низкая точность и recall для класса "кукуруза", что указывает на то, что модель затрудняется в правильной классификации этой конкретной культуры.

На основе результатов тестирования, учитывая высокую точность и сбалансированность метрик, рекомендуется предпочтение отдать градиентному бустингу в контексте классификации сельскохозяйственных культур по сравнению с наивным байесовским алгоритмом. Вместе, снижение размерности Umap и применение метода градиентного бустинга для классификации культур

растений, будут подвергаться дальнейшему исследованию и модернизации в рамках проекта с целью улучшения производительности, расширения функциональности и адаптации к возможным изменениям в данных.

### 1.3 Десктопное приложение для интерактивного анализа временных рядов NDVI и тестирования моделей машинного обучения

Несмотря на наличие различных программ, таких как QGIS, DBeaver и других, предназначенных для работы с географической информационной системой, было принято решение разработать собственное приложение на основе фреймворка Qt. Это решение обусловлено потребностью в инструменте, специализированном на быстром и удобном подключении к существующим данным в ГИС и их анализе.

Пользователь имеет возможность установить соединение с удаленным сервером, на котором хранятся данные географической информационной системы (ГИС), и использовать инструменты для анализа этих данных.

Приложение состоит из четырех основных элементов:

1. Главное окно: в этом окне пользователь может получить основную информацию о ГИС, такую как общая статистика данных.
2. Окно классификации культур с помощью ML: здесь пользователь может использовать ранее созданные модели машинного обучения для классификации сельскохозяйственных культур на основе данных конкретного поля.
3. Окно визуализации временных рядов NDVI: в этом окне пользователь может визуализировать временные ряды показателей NDVI для различных участков или культур. Он может анализировать изменения во времени и выявлять тенденции с помощью графика, либо же сохранить его в формате png или jpg.

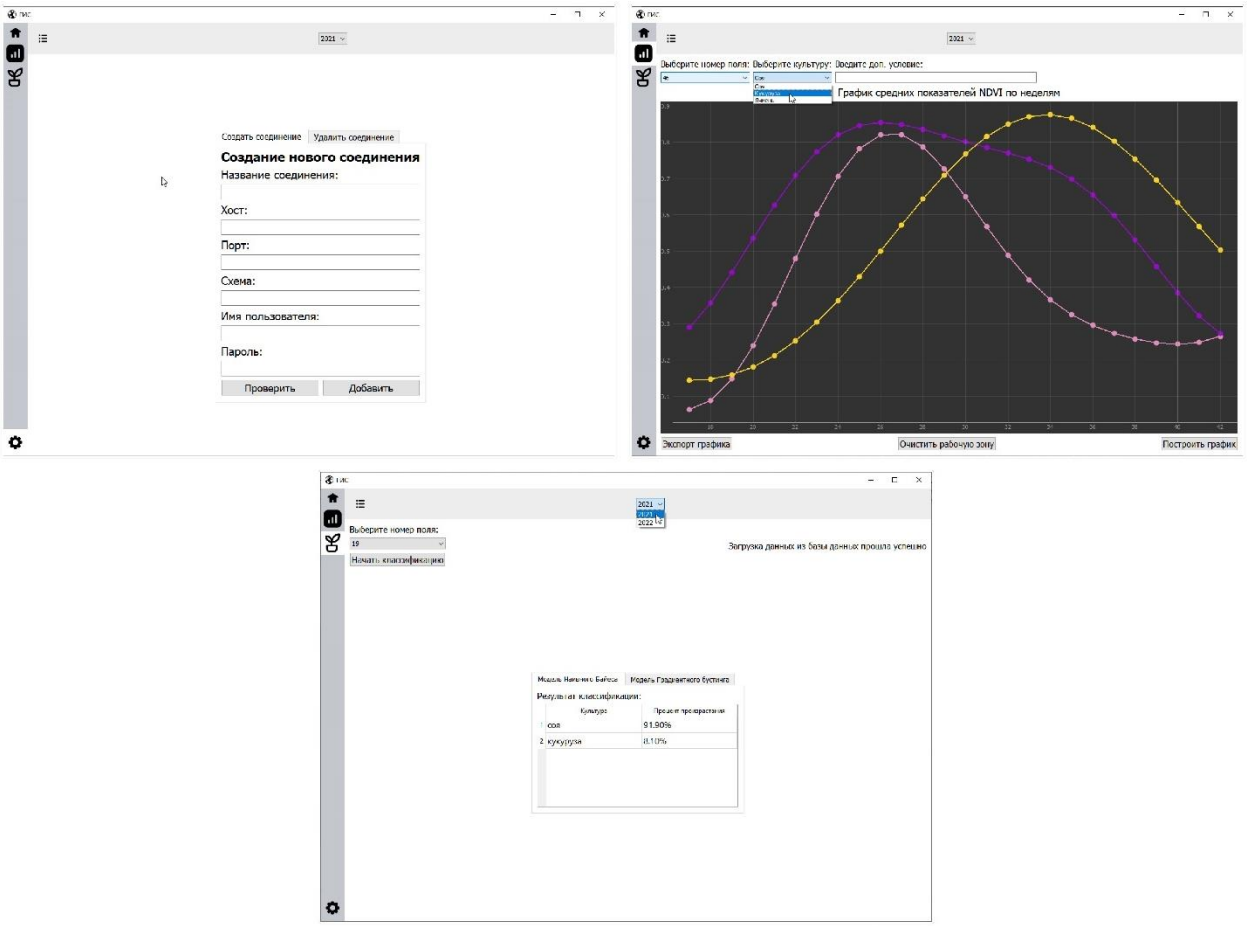


*Рис. 11.* окно настроек, окно визуализации данных, окно классификации

В рамках данного приложения предоставляется функция calculatePlotAvg(), которая позволяет пользователю вычислить средние значения показателя NDVI для заданного поля или культуры растений на протяжении всего временного ряда. Пользователь может выбрать соответствующие параметры, такие как номер поля в базе данных или конкретную культуру, для фильтрации данных, которые в последующем будут использоваться для визуального представления. Результатом выполнения функции является построение графика зависимости среднего значения показателя NDVI от времени для всех точек, соответствующих выбранным параметрам. Это позволяет значительно упростить анализ большого объема данных, поскольку количество значений NDVI может достигать до 30000 точек, каждая из которых имеет 26 вычисленных показателей NDVI. Данный функционал полезен для анализа изменений состояния различных сельскохозяйственных культур, выявления

выбросов или сравнения состояния растений на разных полях или при различных культурах.

В окне визуализации пользователь получает возможность выбрать номер конкретного поля из списка доступных полей. После выбора поля пользователь запускает классификацию точек данного поля с использованием двух ранее созданных моделей машинного обучения. После завершения классификации пользователю выводится результат в виде процентного соотношения для каждой сельскохозяйственной культуры. Это соотношение показывает, к каким культурам растений модели относят растения, выращиваемые на данном участке земли, а также с какой вероятностью.

Особенностью окна визуализации является возможность сравнения результатов, полученных от двух различных моделей. Пользователь может видеть различия в ответах каждой из построенных моделей, что позволяет оценить их эффективность и сравнить их точность классификации для данного участка земли. Такое сравнение помогает пользователям принимать более информированные решения на основе результатов, полученных от различных моделей машинного обучения.

Для создания данного Qt приложения на Python были использованы следующие библиотеки:

- PyQt5 - библиотека для создания графического пользовательского интерфейса.
- sqlite3 - встроенный модуль Python для работы с базой данных SQLite.
- psycopg2 - библиотека для работы с базами данных PostgreSQL.
- Pyqtgraph - библиотека для визуализации данных в реальном времени в PyQt приложениях.
- Numpy - библиотека для работы с многомерными массивами данных.
- Pandas - библиотека для обработки и анализа данных, предоставляющая структуры данных и операции для манипуляции данными.

- Umap - библиотека, реализующая алгоритм UMAP для снижения размерности данных.
- Joblib - библиотека для сохранения и загрузки моделей машинного обучения в Python.

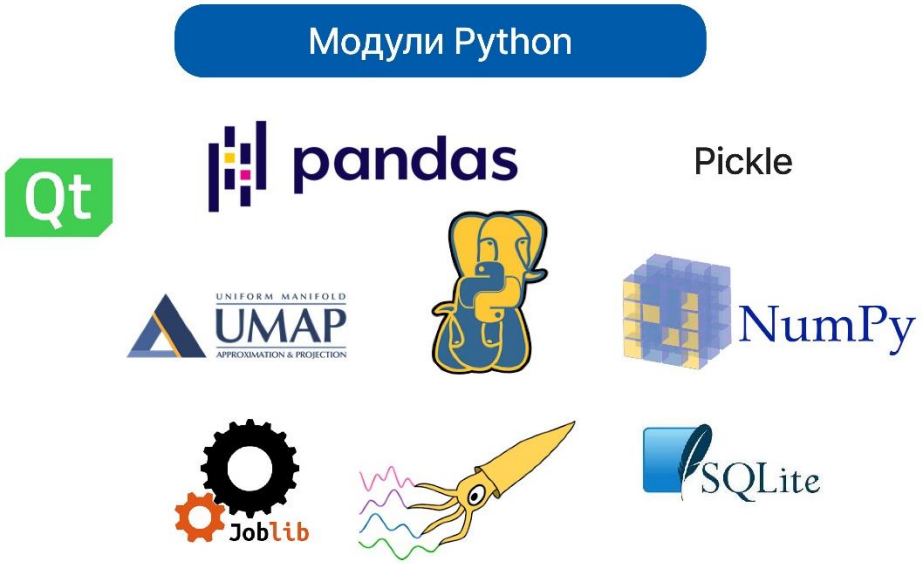


*Рис. 10.* Модули Python, использованные для создания программы

В результате было разработано дополнительное Qt приложение, предназначенное для работы с базами данных, визуализации данных и машинного обучения. Это приложение позволяет пользователям легко подключаться к удаленному серверу, анализировать данные ГИС и классифицировать сельскохозяйственные культуры с использованием моделей машинного обучения.

## Заключение

В ходе выполнения проекта была разработана ГИС для обработки и хранения данных дистанционного зондирования и результатов классификации сельскохозяйственных культур Белогорского района Амурской области.

Создана оптимизированная для эффективного хранения и управления большим объемом геопространственной информации структура, основой для которой создана БД, включающая в себя реализованные инструменты для фильтрации данных, которые позволяют пользователям выделять и анализировать интересующие участки и культуры. Визуализация временных рядов различных вегетационных показателей упрощает восприятие информации, делая анализ более доступным за более короткое время.

В итоге, проведена работа по созданию и оптимизации моделей машинного обучения для классификации сельскохозяйственных культур. В рамках проекта использовались два подхода: сжатие размерности данных с помощью метода UMAP и применение классификаторов, таких как градиентный бустинг и наивный байесовский метод Гаусса. Градиентный бустинг продемонстрировал более высокую точность и F1-score по сравнению с наивным байесовским методом. Таким образом, для дальнейшей работы в проекте был выбран градиентный бустинг в качестве основной модели для классификации сельскохозяйственных культур.

Помимо этого, было создано Qt-приложение, которое обеспечивает пользователей дополнительными инструментами для работы с этой ГИС. Приложение позволяет легко подключаться к удаленному серверу и анализировать данные. В приложении реализована возможность классификации сельскохозяйственных культур с использованием моделей машинного обучения.

Реализованная ГИС позволяет систематизировать и автоматизировать ряд процессов, связанных с мониторингом и управлением сельскохозяйственными угодьями. Например, агрономы могут использовать ГИС для анализа земельного фонда определенного района РФ, определения наилучших участков для

конкретных посевов и планирования ротаций; ГИС может служить платформой для обмена знаниями и информацией между учеными; владельцы полей и агрономы могут использовать модель классификации культур растений для сверки планируемых и фактических культур, что позволяет оперативно выявлять расхождения между планами, а визуализация временных рядов NDVI может помочь в раннем выявлении признаков заболеваний или вредителей с помощью анализа.

Результаты работы докладывали на научной конференции Far East Math – 2023 [3].

К основным направлениям дальнейшего развития ГИС относится разработка простого интерфейса: создание простого и интуитивно понятного пользовательского интерфейса для работы с ГИС. Разработка программного обеспечения, которое позволит пользователям без технических навыков эффективно взаимодействовать с геопространственными данными.

Материалы проекта и документация доступны на следующих платформах:

- GitHub: https://github.com/sprittzer/GIS
- Яндекс Диск: https://disk.yandex.ru/d/9gvFA9b56jd_2A

## Список использованных источников

1. *Буланов К.А., Денисов П.В., Лупян Е.А., Мартьянов А.С., Середа И.И., Трошко К.А., Толпин В.А., Барталев С.А., Хвостиков С.А.* Блок работы с данными дистанционного зондирования Земли Единой федеральной информационной системы о землях сельскохозяйственного назначения // Современные проблемы дистанционного зондирования Земли из космоса, 2019. Т. 16. № 3. С. 171-182.

2. *Илларионова Л.В., Степанов А.С., Фомина Е.А.* Распознавание и классификация посевов сельскохозяйственных культур Хабаровского края с использованием NDVI и EVI // Современные проблемы дистанционного зондирования Земли из космоса, 2023, 20(2), страницы 155–165.

3. *Заварыкина, Т. Д.* Разработка геоинформационной базы данных с использованием POSTGRESQL для хранения данных спутникового мониторинга сельскохозяйственных полей Белогорского района Амурской области / Т. Д. Заварыкина, Л. В. Илларионова // Far East Math - 2023: Материалы национальной научной конференции, Хабаровск, 04–09 декабря 2023 года. – Хабаровск: Тихоокеанский государственный университет, 2024. – С. 91-95. – EDN SLYMPE.