**Платформа автоматизации построения**

**моделей технологических и**

**бизнес-процессов на основе сетевых**

**структур и данных измерений**

описание применения

Санкт-Петербург, 2021 г.

**АННОТАЦИЯ**

Документ содержит описание применения платформы автоматизации построения моделей технологических и бизнес-процессов на основе сетевых структур и данных измерений, разработанной в рамках реализации программы Национального центра когнитивных разработок и договора о предоставлении гранта на государственную поддержку центров Национальной технологической инициативы на базе образовательных организаций высшего образования и научных организаций № 8/1251/2019 от 15.08.2019.

В документе представлено назначение платформы автоматизации построения моделей технологических и бизнес-процессов на основе сетевых структур и данных измерений, подробно описаны условия и порядок выполнения отдельных проектов, а также предусмотренные входные и выходные файлы.

**ОГЛАВЛЕНИЕ**

[1. Описание задачи………………………………………………………………………………………………………………………..4](#Описание_задачи)

[2. Условия выполнения программы………………………………………………………………………………………………4](#Условия_выполнения_программы)

[3. Методы решения задачи ………………………………………………….……………………………………………………….4](#Методы_решения_задачи)

[3.1. Классификация видов Ирисов (проект «Калссификация»)……………….……..](#Классификация)5

[3.2. Подбор параметров модели, построенной на данных «Ирисы» (проект «Hyperparameter optimization»)………….……………………………………………………………16](#Подюор_параметров_модели)

[3.3. Ценообразование инновационных вагонов (проект «Ценообразование вагонов»)………….……..…………………………….………………..……..………………………](#Ценообразование_вагонов)28

[3.4. Временной ряд (проект «Временной ряд»)……………………….. …………………](#Временной_ряд)43

[3.5. Оценка моделей (проект «Models evaluating»)………………………………………52](#Оценка_моделей)

[4. Входные и выходные данные …………………………………………………………………………………………………](#Входные_и_выходные_данные)61

[4.1. Сведения о входных данных ………………………………………..…………………………](#Сведения_о_входных_данных)61

[4.1.1. Данные, загружаемые в проекты ………………………………..…………………](#Данные_загружаемые_в_проекты)61

[4.1.2. Создание пользовательских моделей …………………………..……………….](#Создание_пользовательских_моделей)63

[4.2. Сведения о выходных данных …………………………………..……………………………7](#Сведения_о_выходных_данных)8

1. **Описание задачи**

Программа предназначена для автоматизации разработки технологических и бизнес-процессов крупных производственных предприятий, требующих для своего эффективного функционирования мониторинга деятельности на основе объективных данных, анализа узких мест и экономических потенциалов, оптимизации планирования и максимизации экономических показателей. Основная функциональность программы включает:

* Быстрое прототипирование и создание моделей технологических и бизнес процессов на основе данных измерений (включая модели машинного обучения и искусственного интеллекта)
* Наглядное представление моделей и данных в форме сетевых структур
* Унификация базовых процедур и требований к построению моделей
* Визуализация результатов моделирования

Основная задача, решаемая программой – создание, передача и обслуживание наукоемких данных. Данный фреймворк позволяет реализовывать полный цикл анализа данных включающий чтение данных из различных источников, преобразование и фильтрацию, анализ данных, визуализацию и экспорт. Платформа позволяет автоматизировать построение технических моделей и осуществить статистический анализ данных без привлечения дополнительных знаний, связанных с навыками программирования, что обеспечивает эффективную и непрерывную работу компаний, использующих программу.

1. **Условия выполнения программы**

Минимальные требования функционирования клиентской части:

* Операционная система: Windows XP с пакетом обновления 2 или новее, Windows Vista, Windows 7, Windows 8, Windows 10
* Процессор: Intel Pentium 4 / Athlon 64 или более поздней версии с поддержкой SSE2
* Свободное место на диске: 350 Мб
* Оперативная память: 2048 Mб
* На клиентском компьютере должны быть установлены интернет-браузеры Internet Eplorer, Firefox или Chrome актуальных версий

Персональный компьютер пользователя должен иметь доступ к серверу, на котором развернуто программное обеспечение программного комплекса, а также доступ к сети Интернет.

1. **Методы решения задачи**

*Классификация видов Ирисов (проект «Калссификация»)*

В данном примере необходимо решить задачу разбиения набора данных "Ирисы" на классы с использованием метода классификации разработанной программы.

*Подбор параметров модели, построенной на данных «Ирисы» (проект «Hyperparameter optimization»)*

В данном примере необходимо решить задачу повышения качества обученных моделей логистической регрессии и классификации методом случайного леса, созданных на данных «Ирисы», при помощи настройки гиперпараметров моделей, осуществляемой методом подбора параметров разработанной программы.

*Ценообразование инновационных вагонов (проект «Ценообразование вагонов»)*

В данном примере необходимо решить задачу предсказания цен инновационных вагонов на основе данных предсказанных и имеющихся цен типовых и имеющихся цен инновационных вагонов с повторным использованием метода линейной регрессии для предсказания цен типовых вагонов, и для предсказания цен инновационных вагонов с помощью модели гребневой регрессии.

*Временной ряд (проект «Временной ряд»)*

В данном примере необходимо решить задачу прогнозирования изменения значений некоторого показателя исследуемого процесса, собранных в разные моменты времени, с использованием метода анализа временных рядов.

*Оценка моделей (проект «Models evaluating»)*

В данном примере необходимо решить задачу выбора наиболее подходящей модели, а именно выбора модели классификации, кластеризации или линейной регрессии, при помощи перебора решений задачи, предлагаемых всеми доступными моделями, и подбора модели, лучше всего справившейся с поставленной задачей.

Программа решает задачи анализа данных с помощью таких методов и алгоритмов Машинного обучения, как:

* регрессии,
* временные ряды,
* кластеризация,
* классификация,
* подбор параметров модели,
* и т.д.

**3.1.** **Классификация видов Ирисов (проект «Калссификация»)**

Работа метода классификации на примере классификации «Ирисов» включает в себя:

1. Загрузку/выбор данных о характеристиках видов ирисов в разделе «Данные» (рис. 3.1.1).

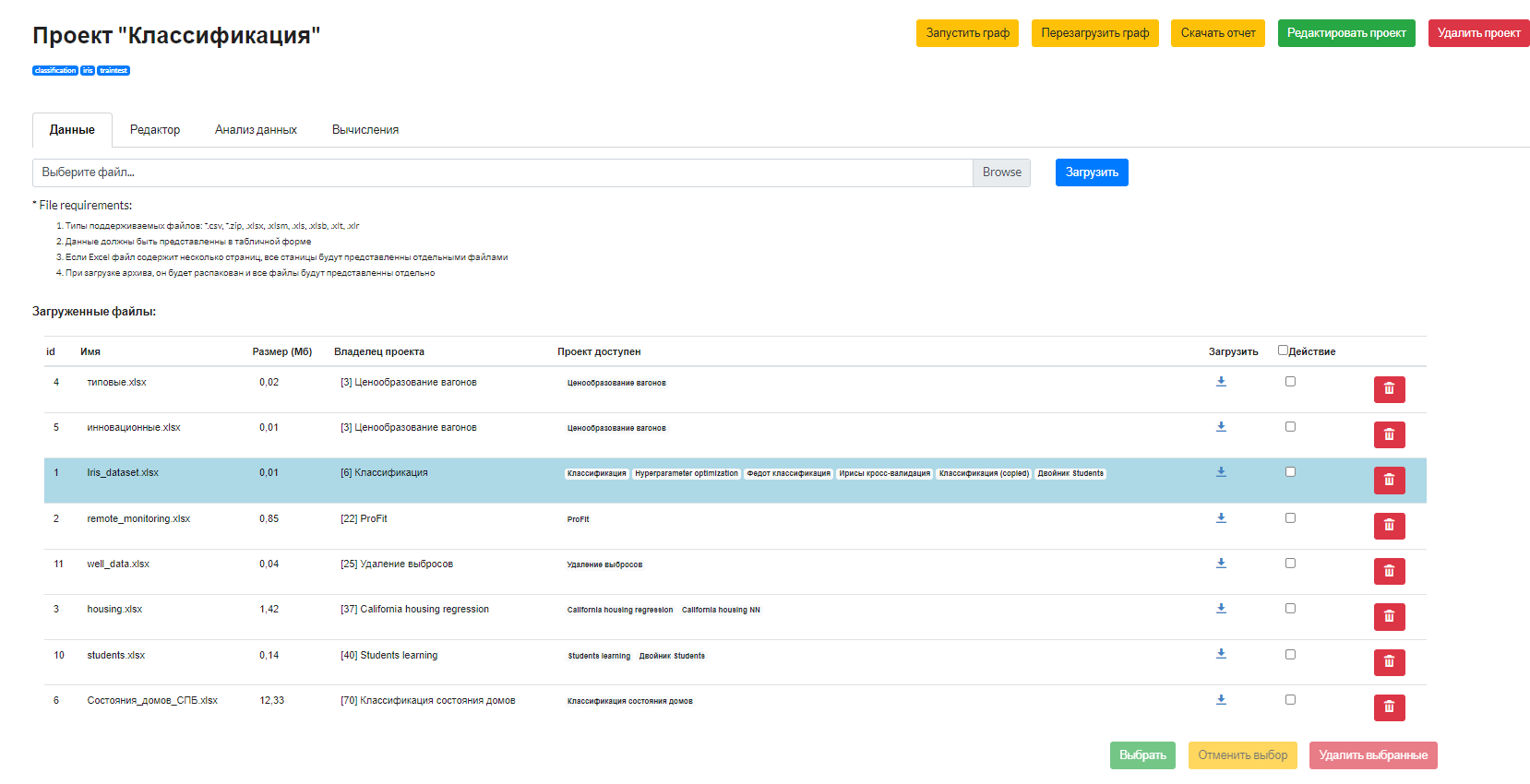


Рисунок 3.1.1 – Загрузка данных «Ирисы» в разделе «Данные»

1. Работу с редактором графов, создаваемых для наглядного представления действий, осуществляемых с данными по ирисам, в разделе «Редактор» (рис. 3.1.2).

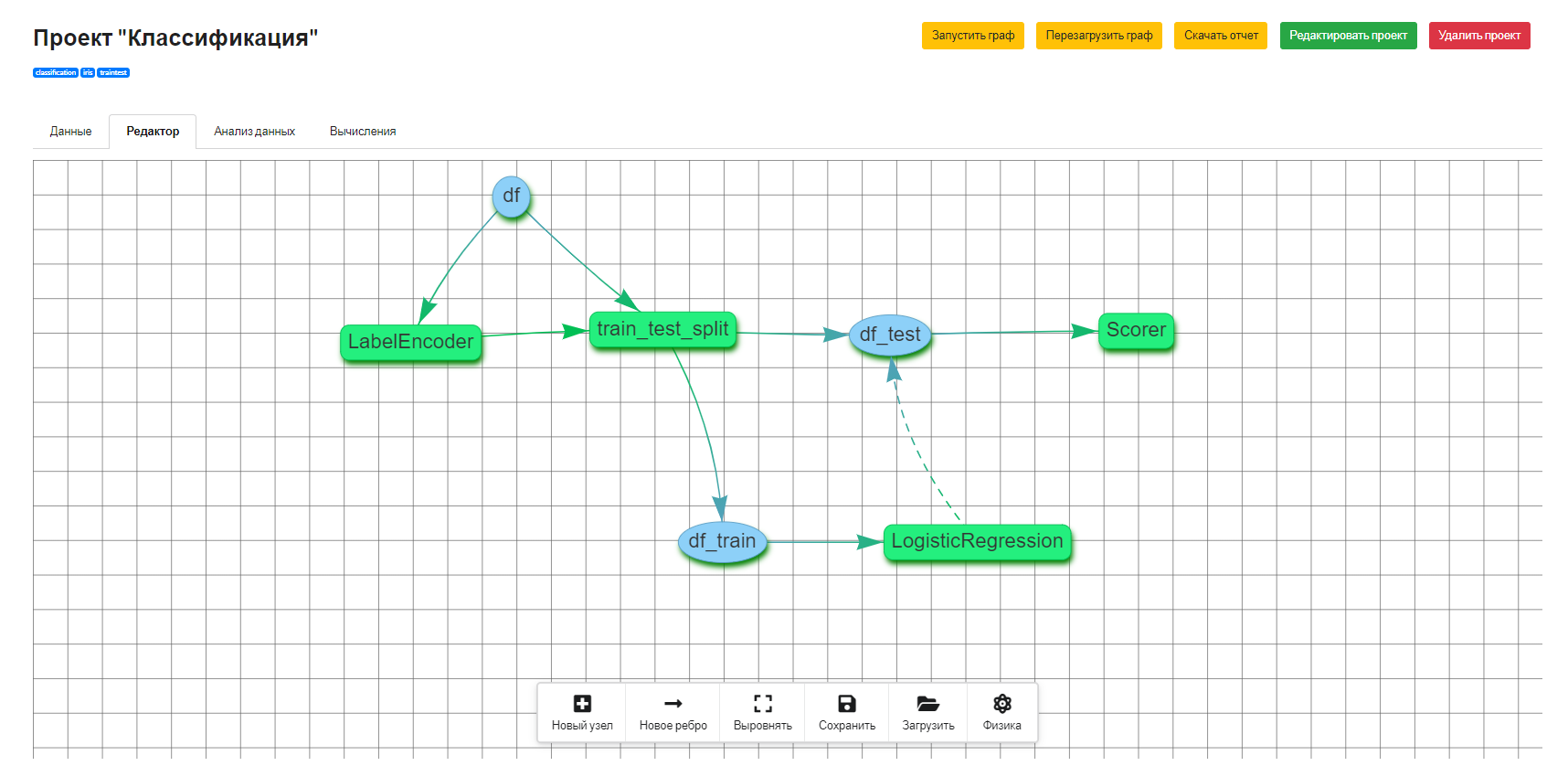


Рисунок 3.1.2 – Работа с редактором графа для данных «Ирисы» в разделе «Редактор»

* 1. Построение графа, а именно добавление узлов и ребер, которые используются для реализации цепи действий, осуществляемых с данными (рис. 3.1.3).

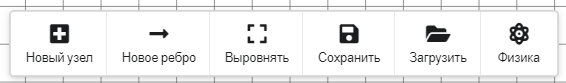


Рисунок 3.1.3 – Меню работы с графами в разделе «Редактор»

Цепь действий для задачи классификации Ирисов:

* Чтение признаков (рис. 3.1.4)

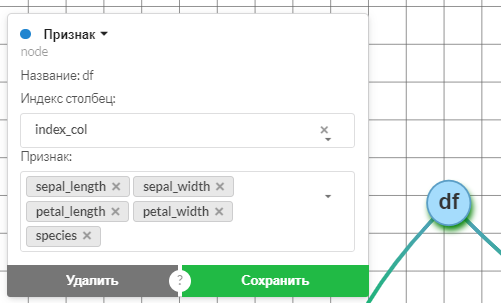


Рисунок 3.1.4 – Узел «Чтение признаков» графа для данных «Ирисы» и его свойства

* Кодировка признаков в числа (рис. 3.1.5)

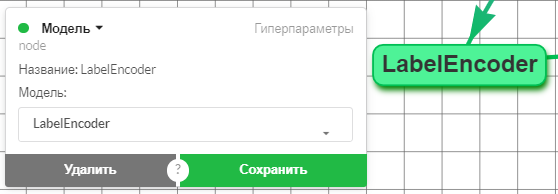


Рисунок 3.1.5 – Узел «Кодировка признаков» графа для данных «Ирисы» и его свойства

* Разбиение выборки (рис. 3.1.6)

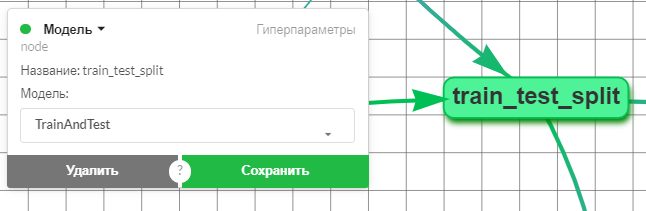


Рисунок 3.1.6 – Узел «Разбиение выборки» графа для данных «Ирисы» и его свойства

* Выделение тренировочной выборки в отдельный массив (рис.3.1.7)

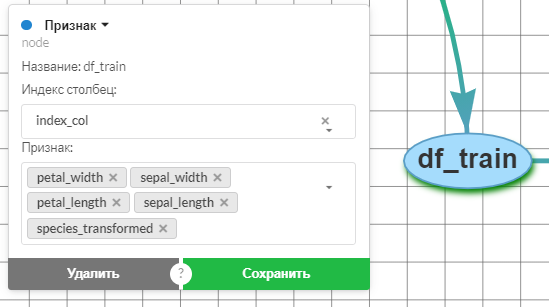


Рисунок 3.1.7 – Узел «Выделение тренировочной выборки в отдельный массив» графа для данных «Ирисы» и его свойства

* Обучение классификатора на тренировочной выборке (рис. 3.1.8)

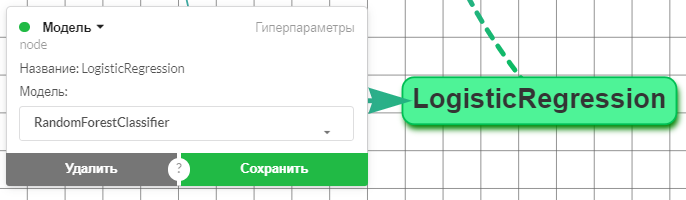


Рисунок 3.1.8 – Узел «Обучение классификатора на тренировочной выборке» графа для данных «Ирисы» и его свойства

* Применение обученного классификатора на тестовой выборке (рис. 3.1.9)

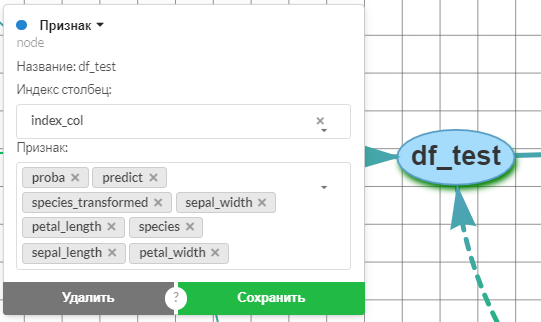


Рисунок 3.1.9 – Узел «Применение обученного классификатора на тестовой выборке» графа для данных «Ирисы» и его свойства

* Отображение результатов применения классификатора (рис. 3.1.10)

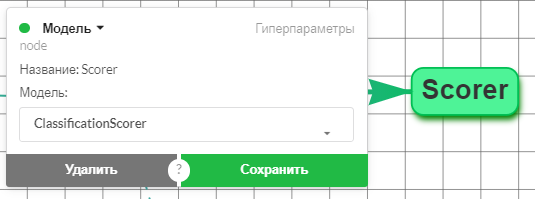


Рисунок 3.1.10 – Узел «Отображение результатов применения классификатора» графа для данных «Ирисы» и его свойства

* 1. Выбор свойств узлов/ребер (для признаков - синие узлы, а для моделей - зеленые узлы). Признаками в примере являются «Перевод результатов кодировки в статус выборки», «Чтение признаков», «Тестовая выборка» и «Тренировочная выборка», а моделями – «Кодировка меток в числа», «Разбиение выборки», «Обучение классификатора на тренировочной выборке» и «Отображение результатов применения классификатора» (рис. 3.1.11).

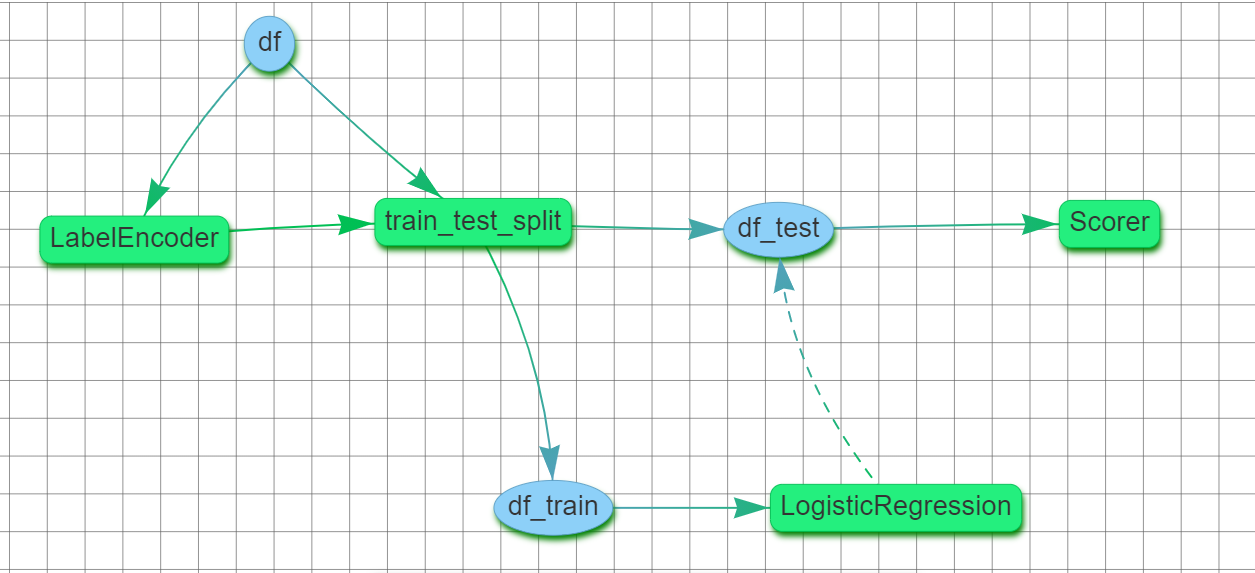


Рисунок 3.1.11 – Граф для данных «Ирисы» в разделе «Редактор»

1. Процесс просмотра входных данных проекта классификации ирисов в разделе «Анализ данных» (рис. 3.1.12-3.1.13).

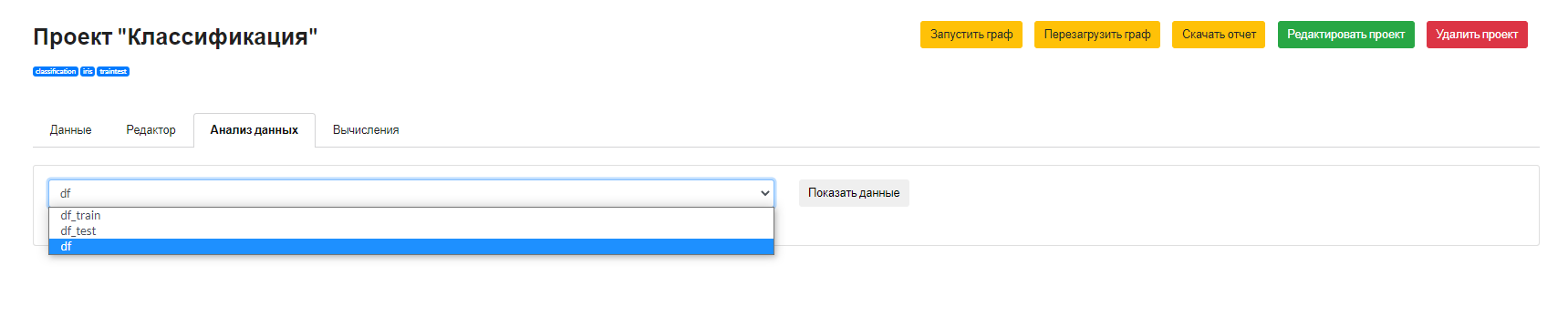


Рисунок 3.1.12 – Раздел «Анализ данных» проекта классификации данных «Ирисы»

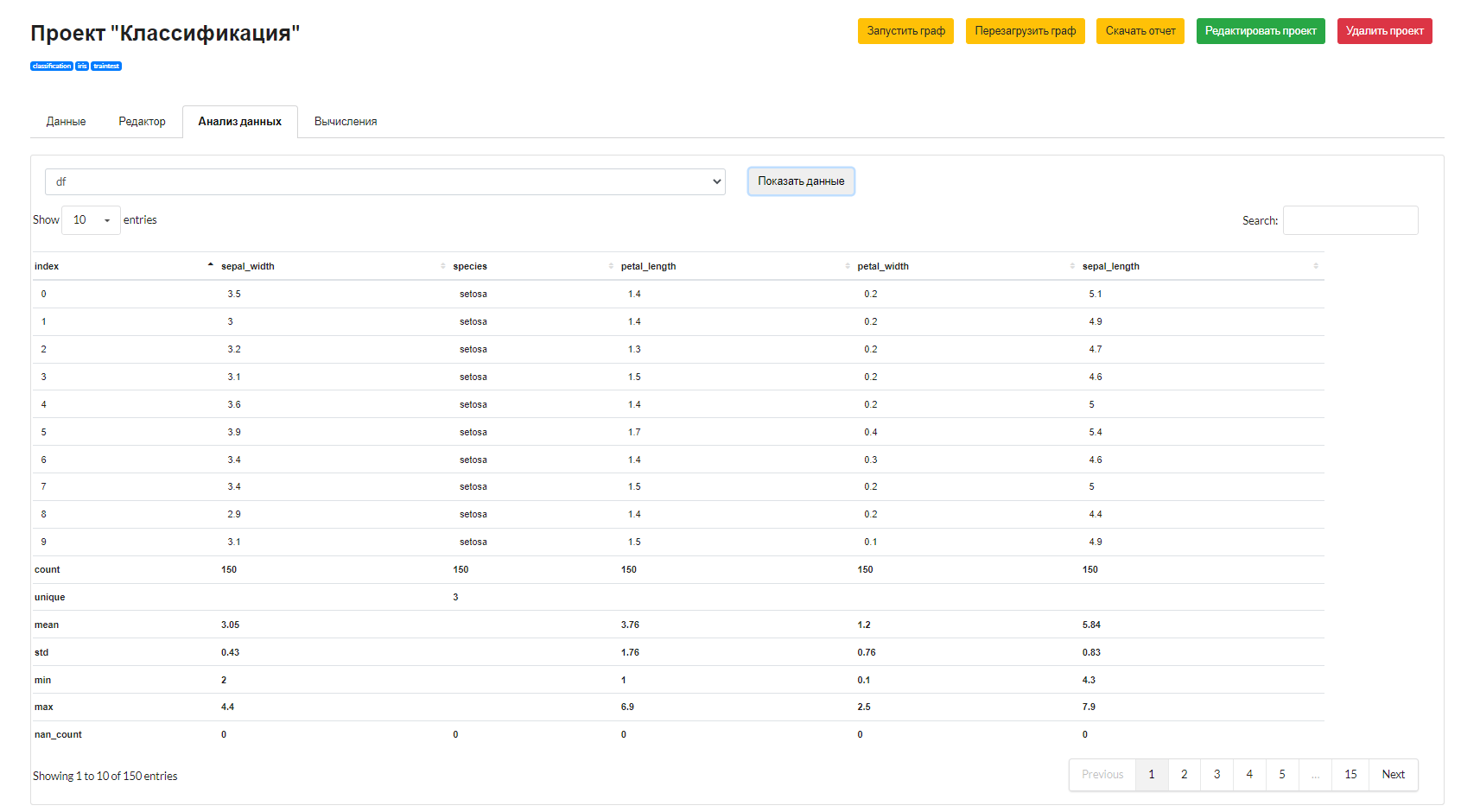


Рисунок 3.1.13 – Просмотр входных данных проекта классификации данных «Ирисы»

1. Процесс вычисления, осуществляемого моделью классификации, в разделе «Вычисления» (рис. 3.1.14).

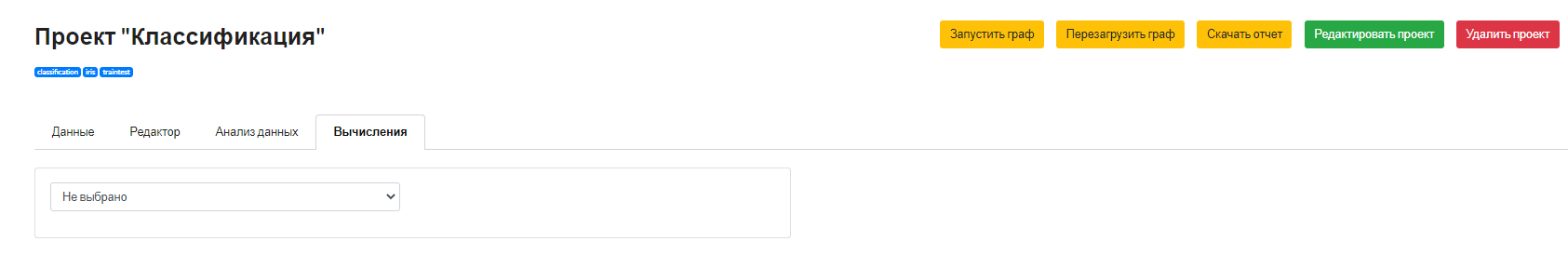


Рисунок 3.1.14 – Раздел «Вычисления» проекта классификации данных «Ирисы»

* 1. Выбор узла с моделью классификации (рис. 3.1.15).

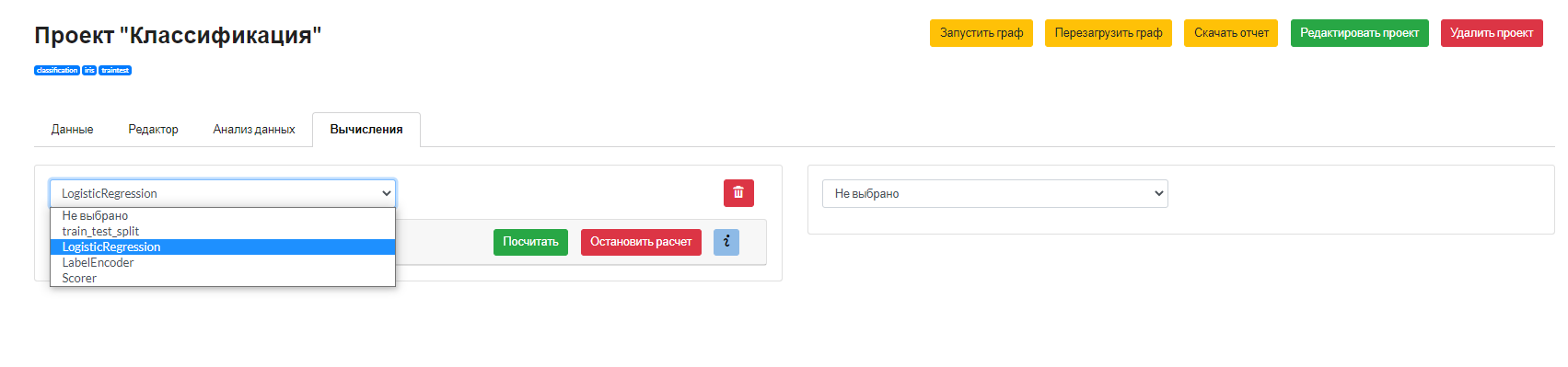


Рисунок 3.1.15 – Выбор узла с моделью классификации в разделе «Вычисления»

* 1. Настройка модели, а именно установка параметров, по которым будет осуществлена классификация ирисов, в меню «Расширенные настройки» (рис. 3.1.16).

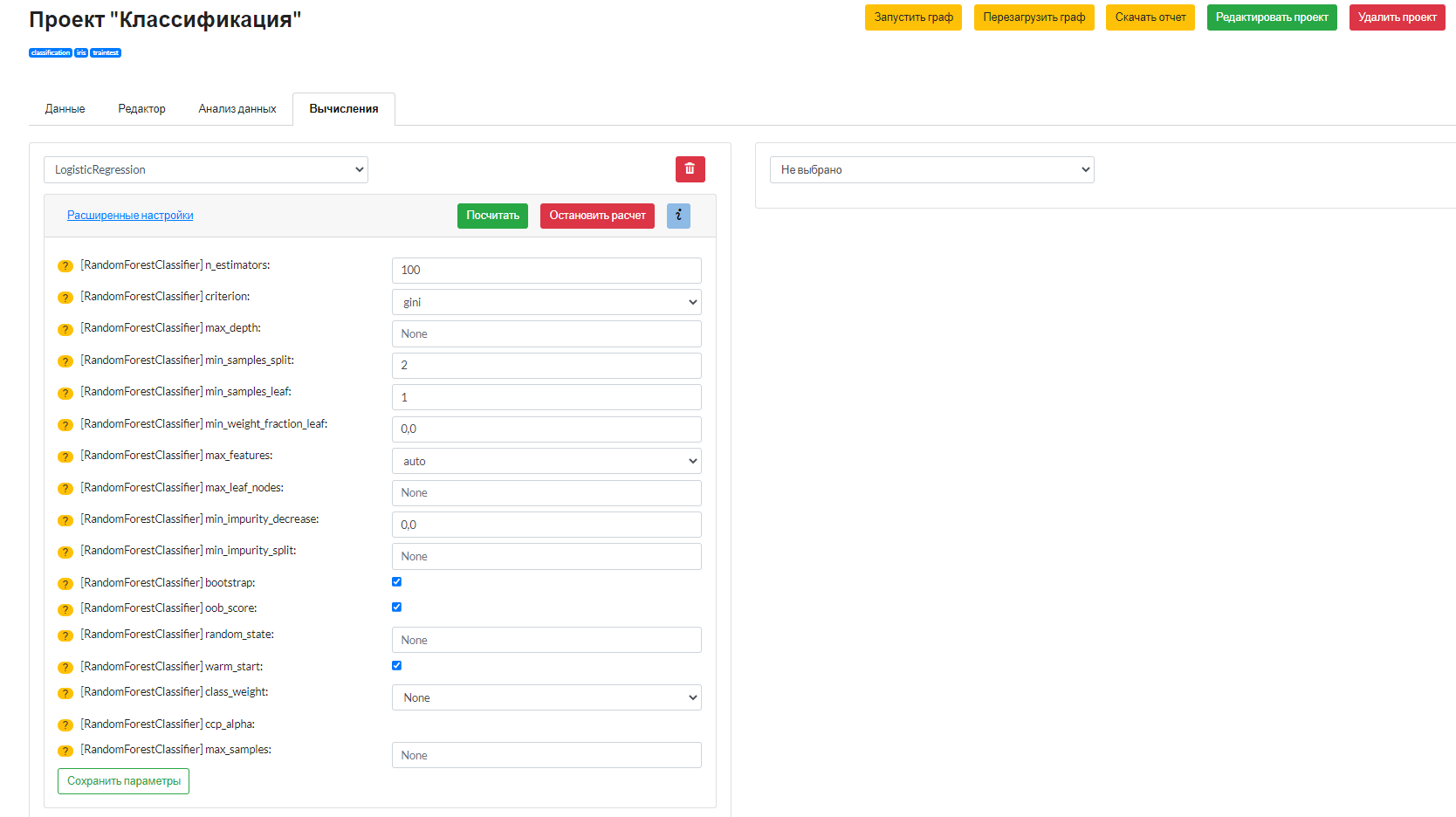


Рисунок 3.1.16 – Меню «Расширенные настройки» проекта классификации данных «Ирисы» в разделе «Вычисления»

* 1. Переход между вкладками с параметрами, полученными в результате работы модели классификации ирисов (рис. 3.1.17).

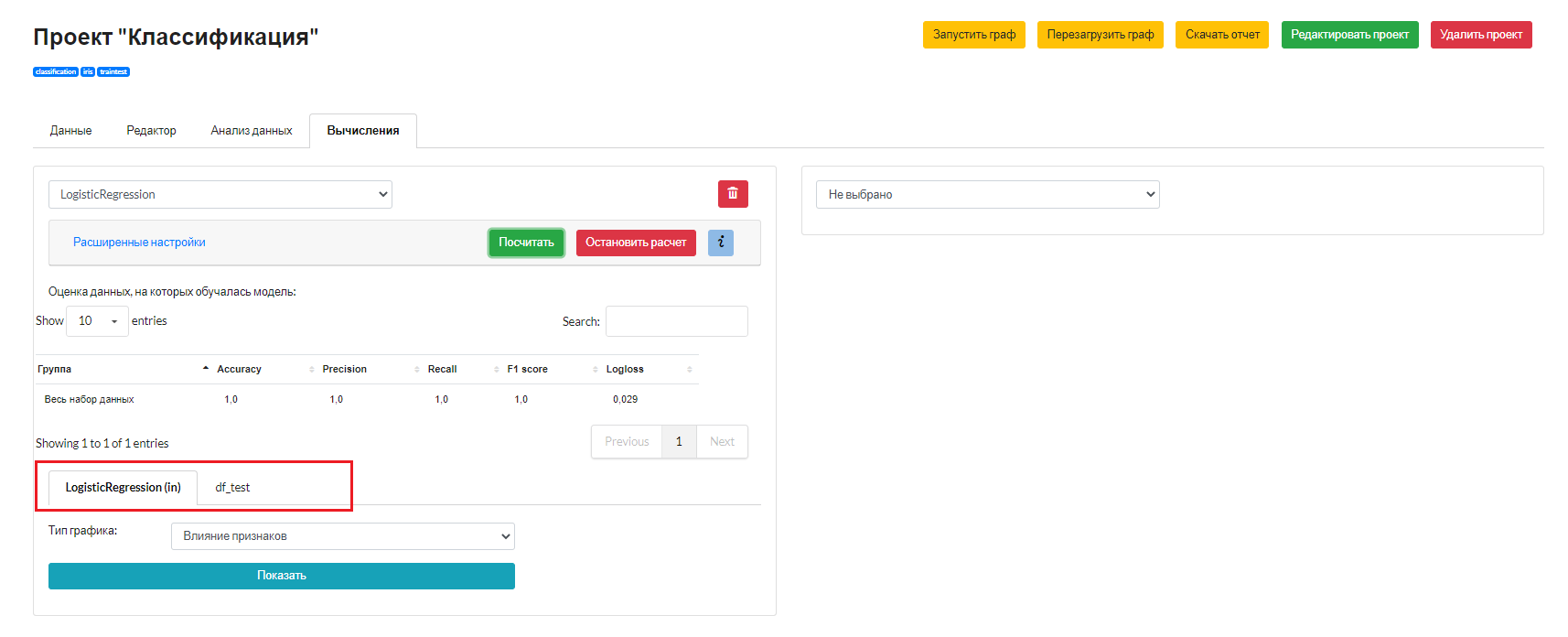


Рисунок 3.1.17 – Выбор вкладки с параметром, полученным в результате работы модели классификации ирисов, в разделе «Вычисления»

* 1. Выбор типа визуализации построения классов ирисов (рис. 3.1.18), типа данных (3.1.19) и его визуализация (рис. 3.1.20).

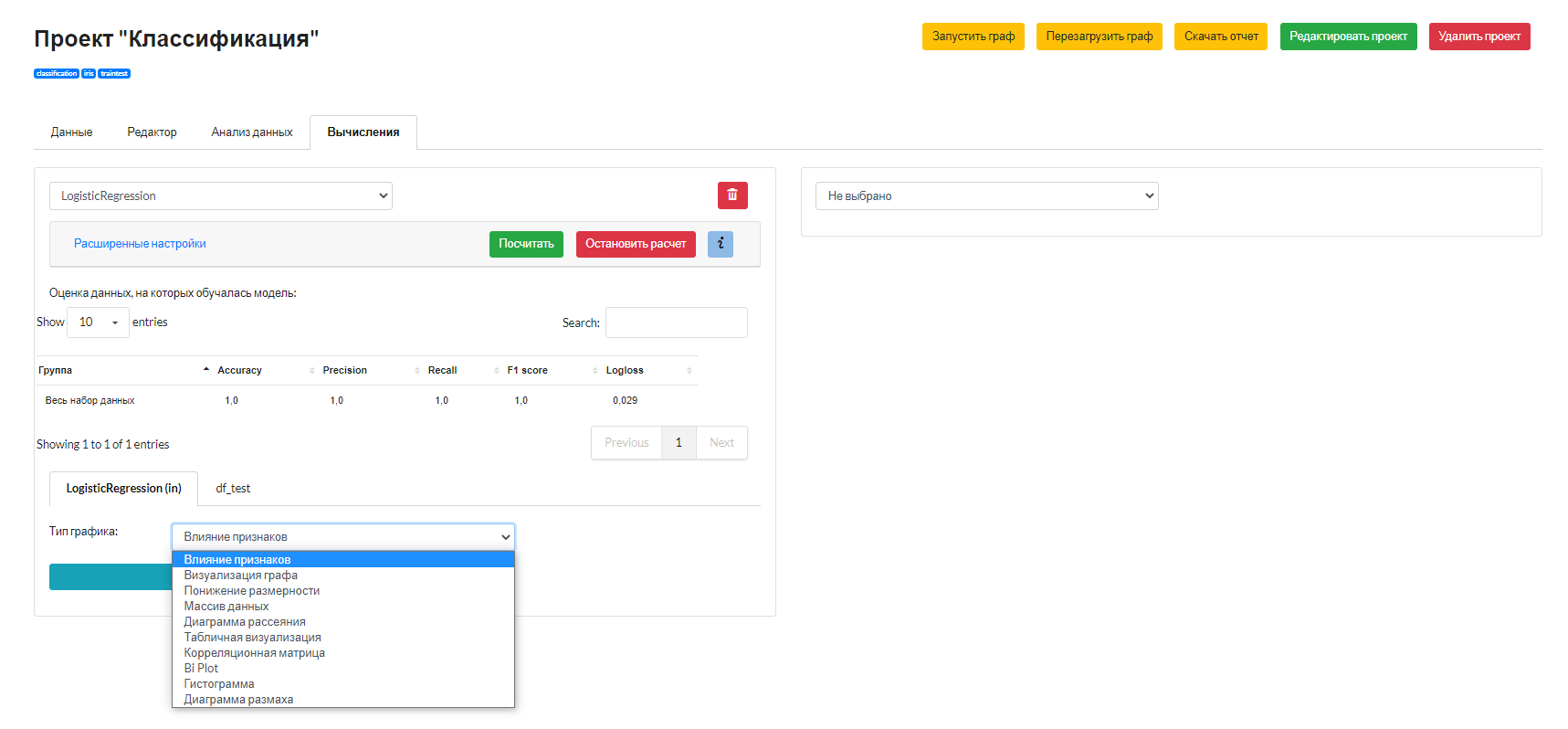


Рисунок 3.1.18 – Выбор типа визуализации результатов работы модели классификации данных «Ирисы» в разделе «Вычисления»

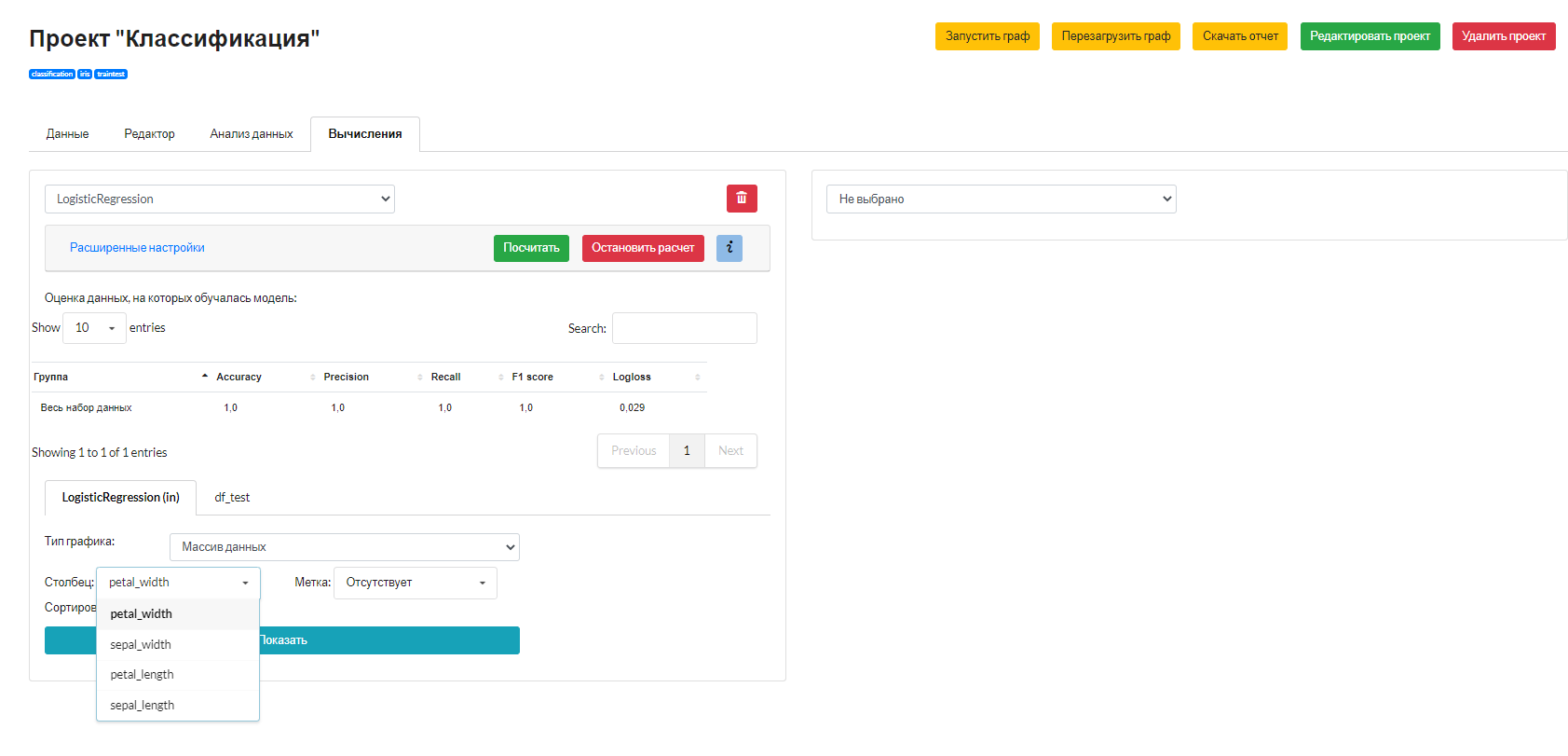


Рисунок 3.1.19 – Выбор типа данных результатов работы модели классификации данных «Ирисы» в разделе «Вычисления»

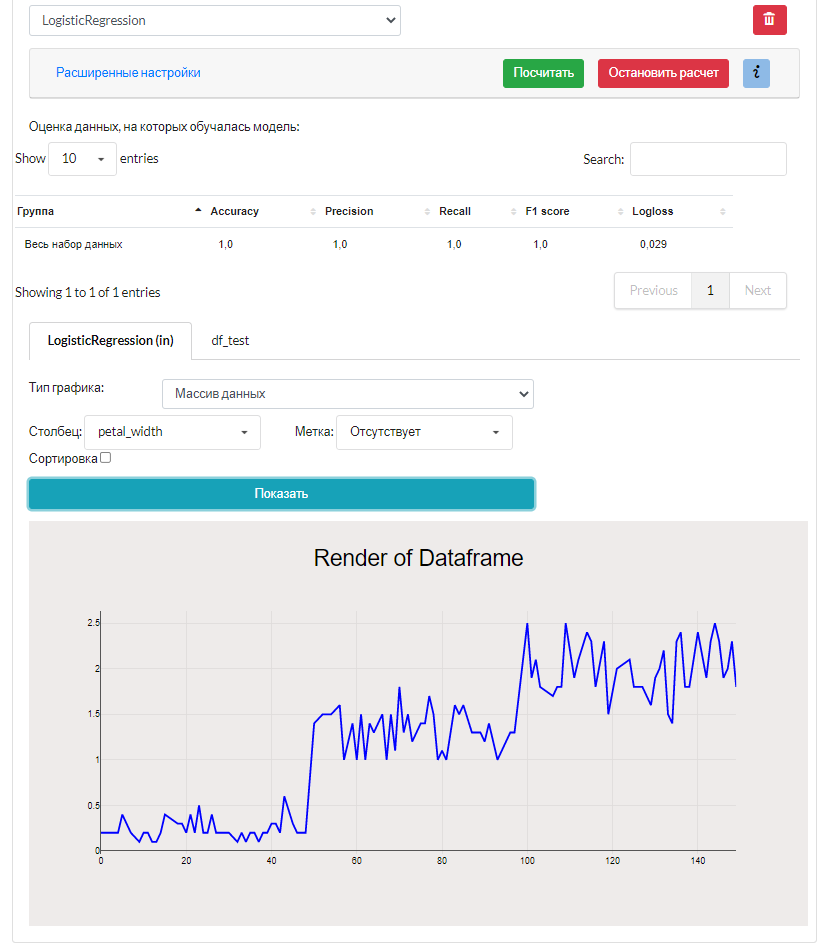


Рисунок 3.1.20 – Визуализация графика результатов работы модели классификации данных «Ирисы» в разделе «Вычисления»

1. Сохранение результатов работы проекта.
   1. Загрузка результатов работы модели, содержащих классифицированные данные ирисов на их виды, на ПК пользователя в разделе «Вычисления» (рис. 3.1.21).

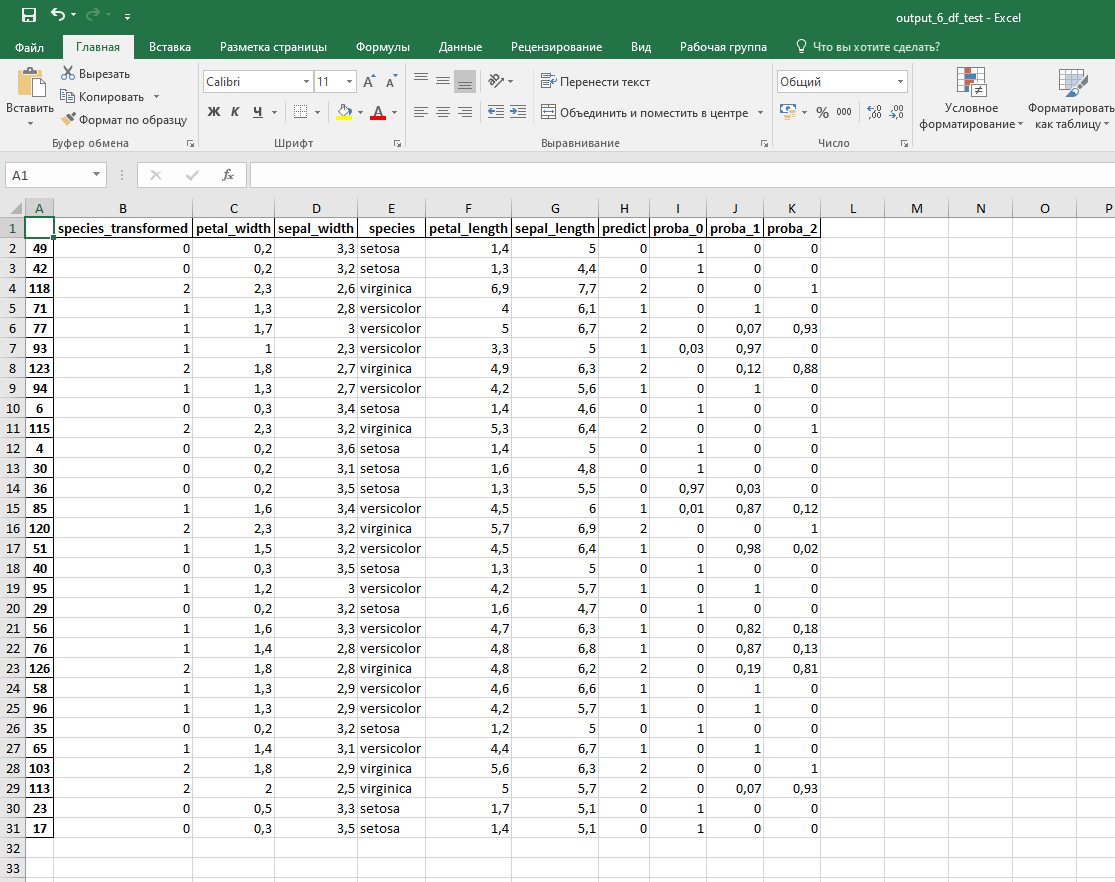


Рисунок 3.1.21 – Файл с результатами работы модели классификации данных «Ирисы» в разделе «Вычисления» на ПК пользователя

* 1. Загрузка отчетов работы модели на ПК пользователя из:
* Раздела «Данные»
* Раздела «Редактор» (рис. 3.1.22)
* Раздела «Анализ данных»
* Раздела «Вычисления»

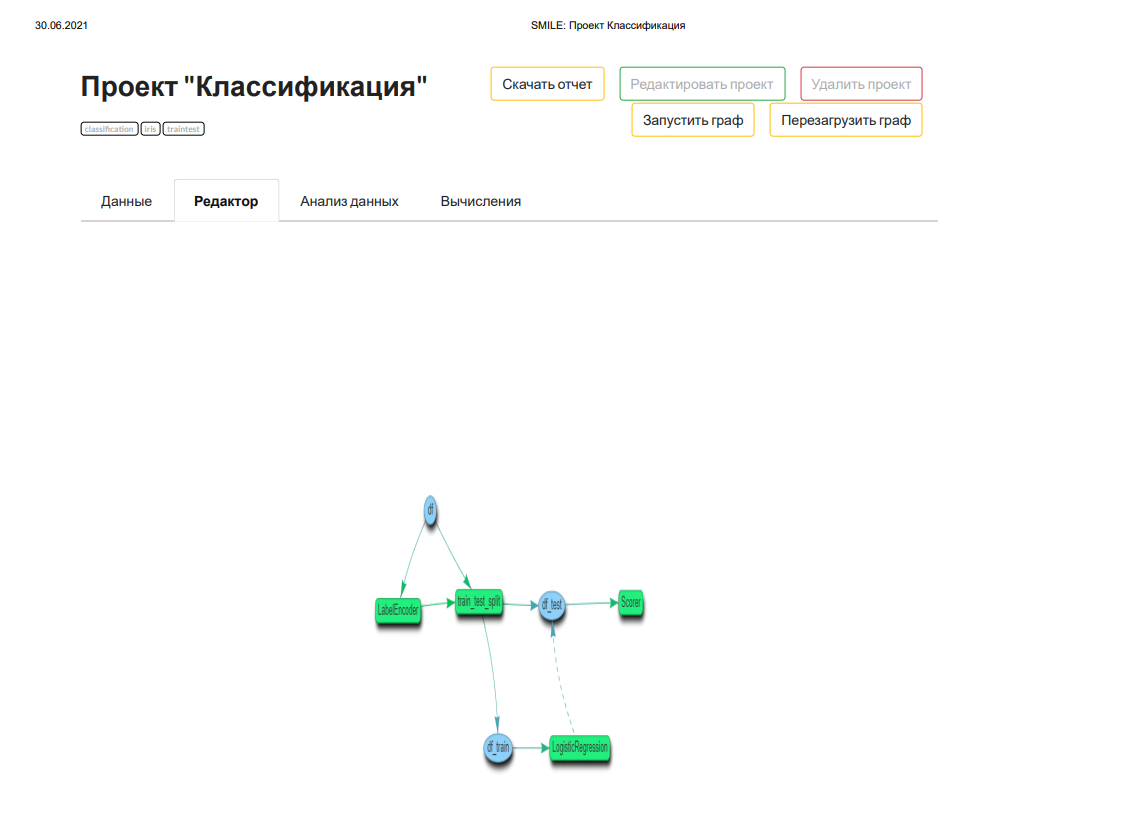


Рисунок 3.1.22 – Отчет раздела «Редактор» проекта классификации ирисов

**3.2. Подбор параметров модели, построенной на данных «Ирисы» (проект «Hyperparameter optimization»)**

Работа метода подбора параметров модели на примере подбора гиперпараметров модели логистической регрессии и классификации методом случайного леса, созданной на данных «Ирисы», включает в себя:

1. Загрузку/выбор данных о характеристиках видов ирисов в разделе «Данные» (рис. 3.2.1).

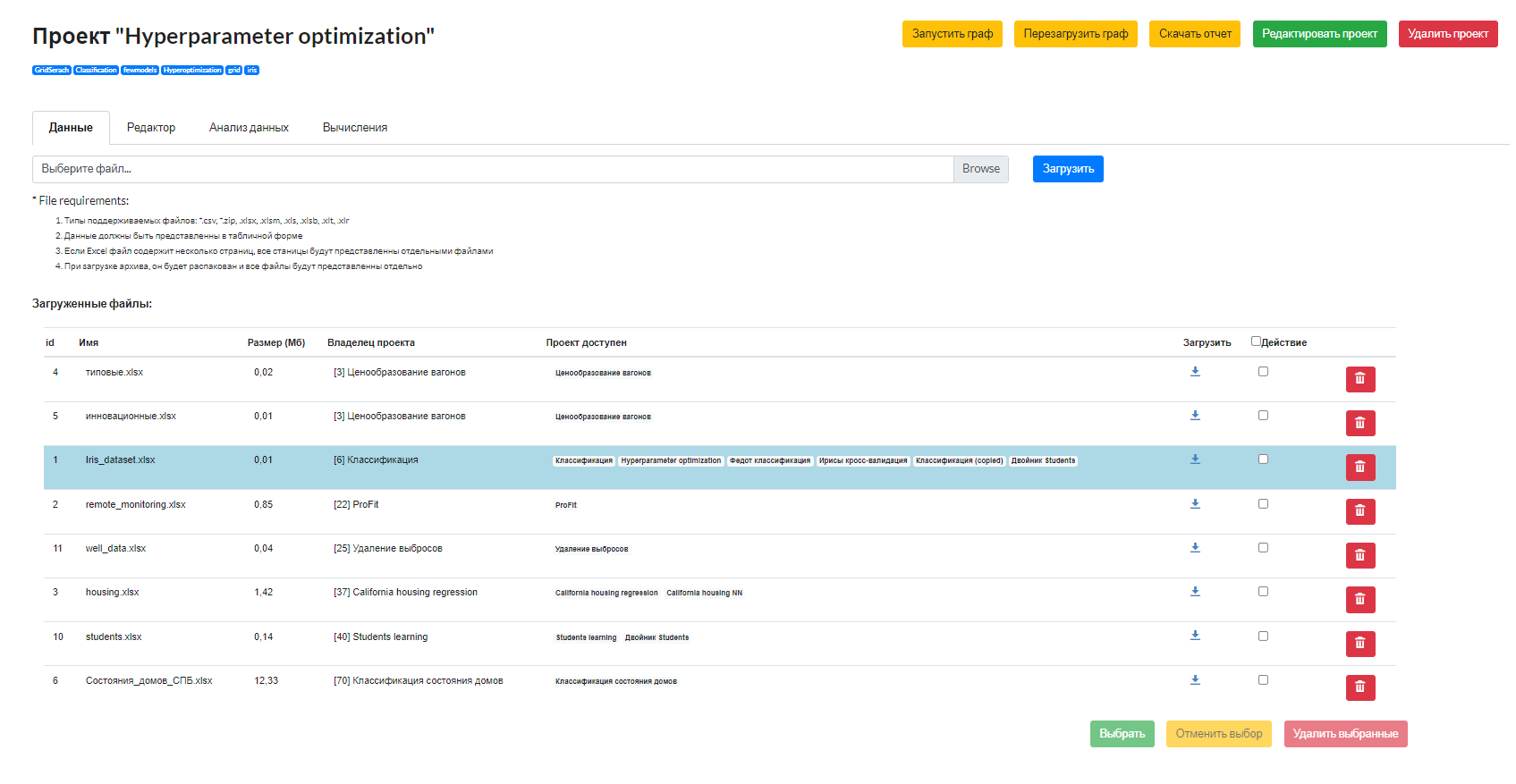


Рисунок 3.2.1 – Загрузка данных «Ирисы» в разделе «Данные»

1. Работу с редактором графов, создаваемых для наглядного представления действий, осуществляемых с данными по ирисам, в разделе «Редактор» (рис. 3.2.2).

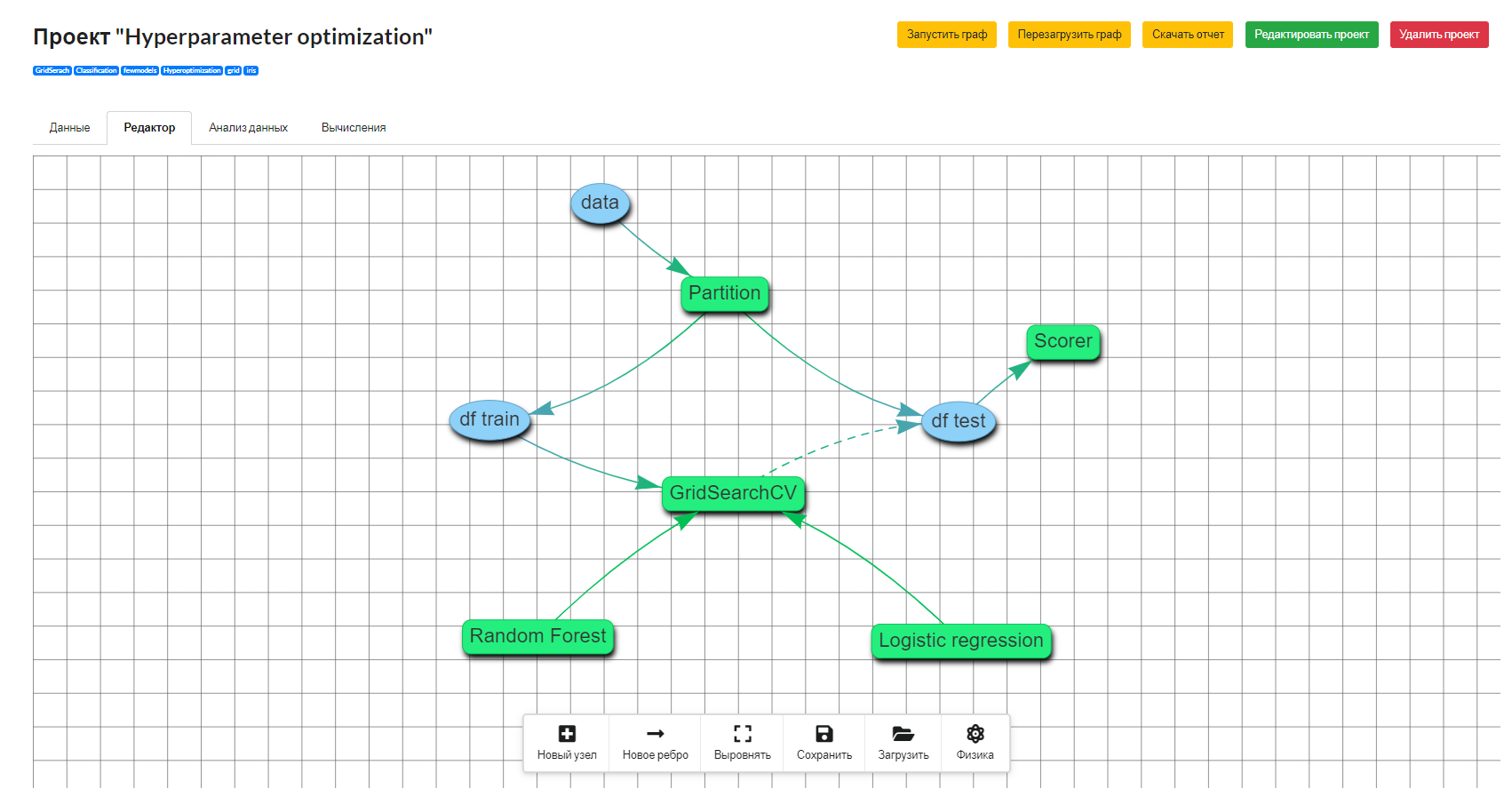


Рисунок 3.2.2 – Работа с редактором графа для данных «Ирисы» в разделе «Редактор»

* 1. Построение графа, а именно добавление узлов и ребер, которые используются для реализации цепи действий, осуществляемых с данными (рис. 3.2.3).

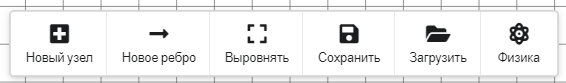


Рисунок 3.2.3 – Меню работы с графами в разделе «Редактор»

Цепь действий для задачи задачи подбора гиперпараметров модели логистической регрессии, созданной на данных «Ирисы»:

* Чтение данных (рис. 3.2.4)

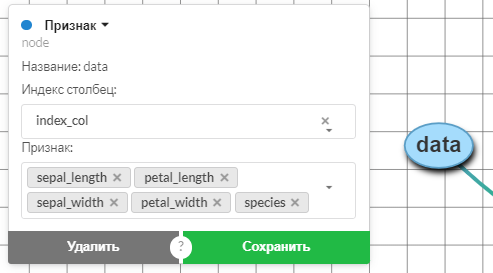


Рисунок 3.2.4 – Узел «Чтение данных» графа для данных «Ирисы» и его свойства

* Модель логистической регрессии, построенная на данных «Ирисы» (рис. 3.2.5)

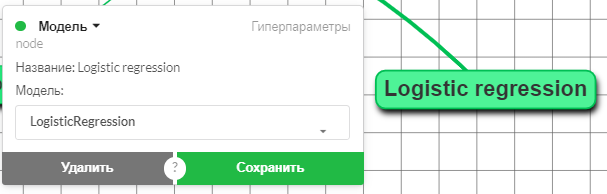


Рисунок 3.2.5 – Узел «Модель логистической регрессии» графа для данных «Ирисы» и его свойства

* Модель классификации методом случайного леса, построенная на данных «Ирисы» (рис. 3.2.6)

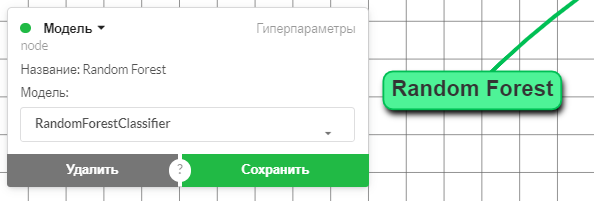


Рисунок 3.2.6 – Узел Модель классификации методом случайного леса» графа для данных «Ирисы» и его свойства

* Разбиение выборки (рис.3.2.7)

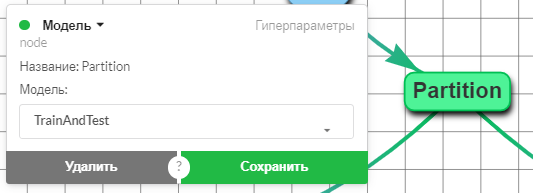


Рисунок 3.2.7 – Узел «Разбиение выборки» графа для данных «Ирисы» и его свойства

* Выделение тренировочной выборки в отдельный массив (рис. 3.2.8)

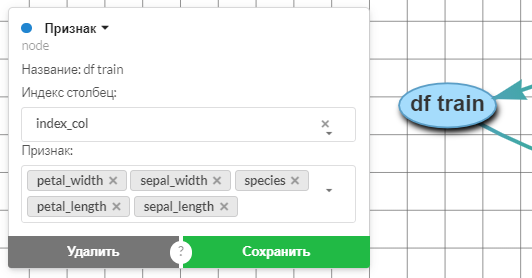


Рисунок 3.2.8 – Узел «Выделение тренировочной выборки в отдельный массив» графа для данных «Ирисы» и его свойства

* Поиск оптимальных параметров модели логистической регрессии, построенной на данных «Ирисы» (рис. 3.2.9)

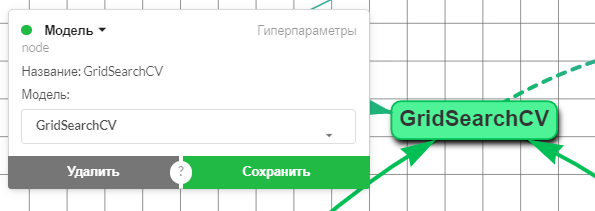


Рисунок 3.2.9 – Узел «Модель GridSearchCV» графа для данных «Ирисы» и его свойства

* Применение поиска оптимальных параметров на тестовой выборке (рис. 3.2.10)

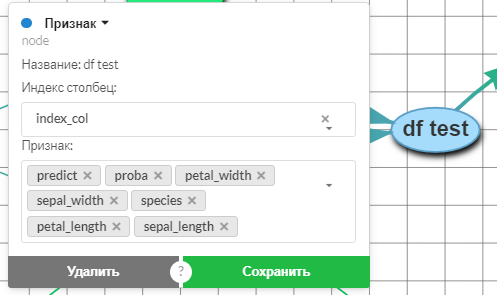


Рисунок 3.2.10 – Узел «Применение поиска оптимальных параметров» графа для данных «Ирисы» и его свойства

* Подсчет меток классификации (рис. 3.2.11)

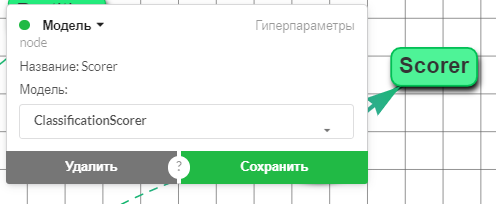


Рисунок 3.2.11 – Узел «Scorer» графа для данных «Ирисы» и его свойства

* 1. Выбор свойств узлов/ребер (для признаков - синие узлы, а для моделей - зеленые узлы). Признаками в примере являются «Чтение данных», «Тестовая выборка» и «Тренировочная выборка», а моделями – «Модель логистической регрессии», «Модель случайного леса», «Применение поиска оптимальных параметров на тренировочной выборке» и «Отображение результатов работы поиска оптимальных параметров» (рис. 3.2.12).

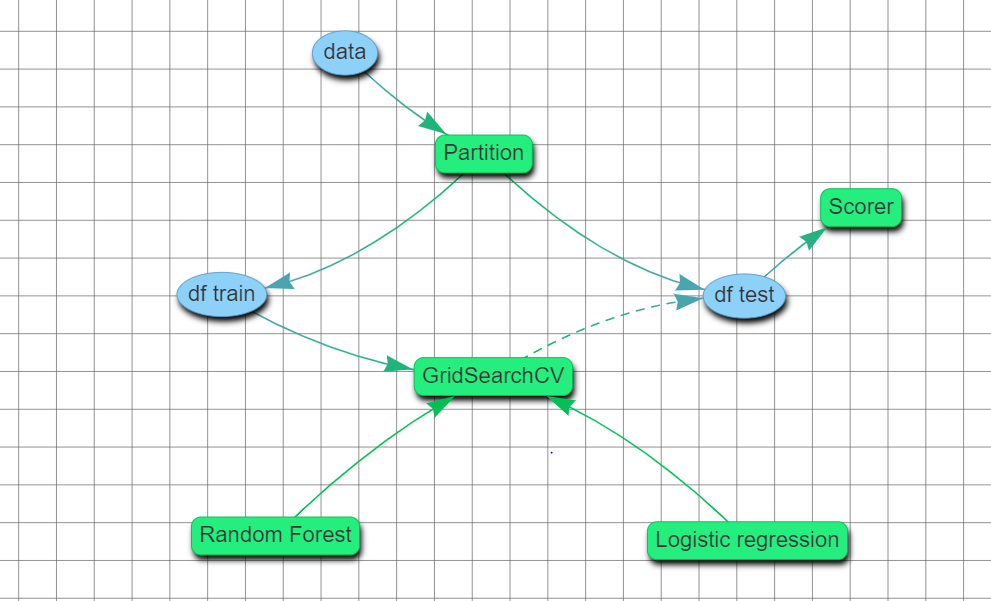


Рисунок 3.2.12 – Граф для данных «Ирисы» в разделе «Редактор»

1. Процесс просмотра входных данных проекта подбора параметров в разделе «Анализ данных» (рис. 3.2.13-3.2.14).

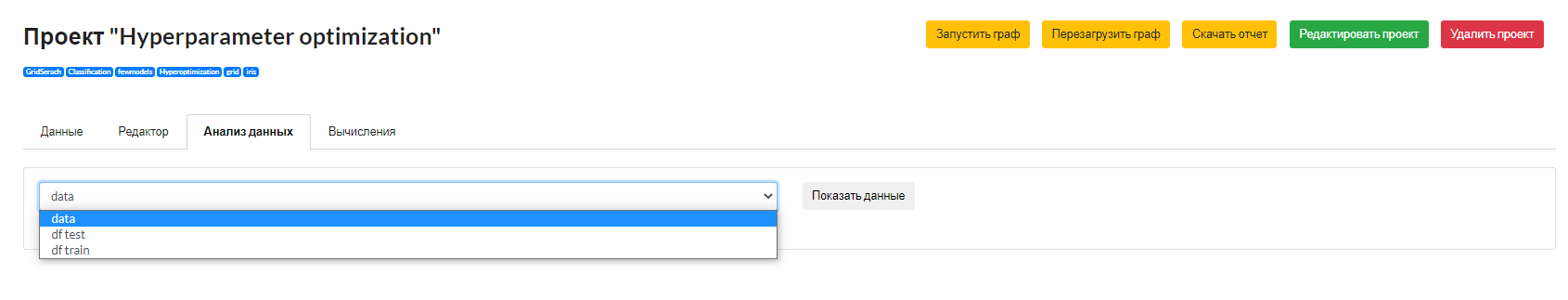


Рисунок 3.2.13 – Раздел «Анализ данных» проекта подбора гиперпараметров моделей данных «Ирисы»

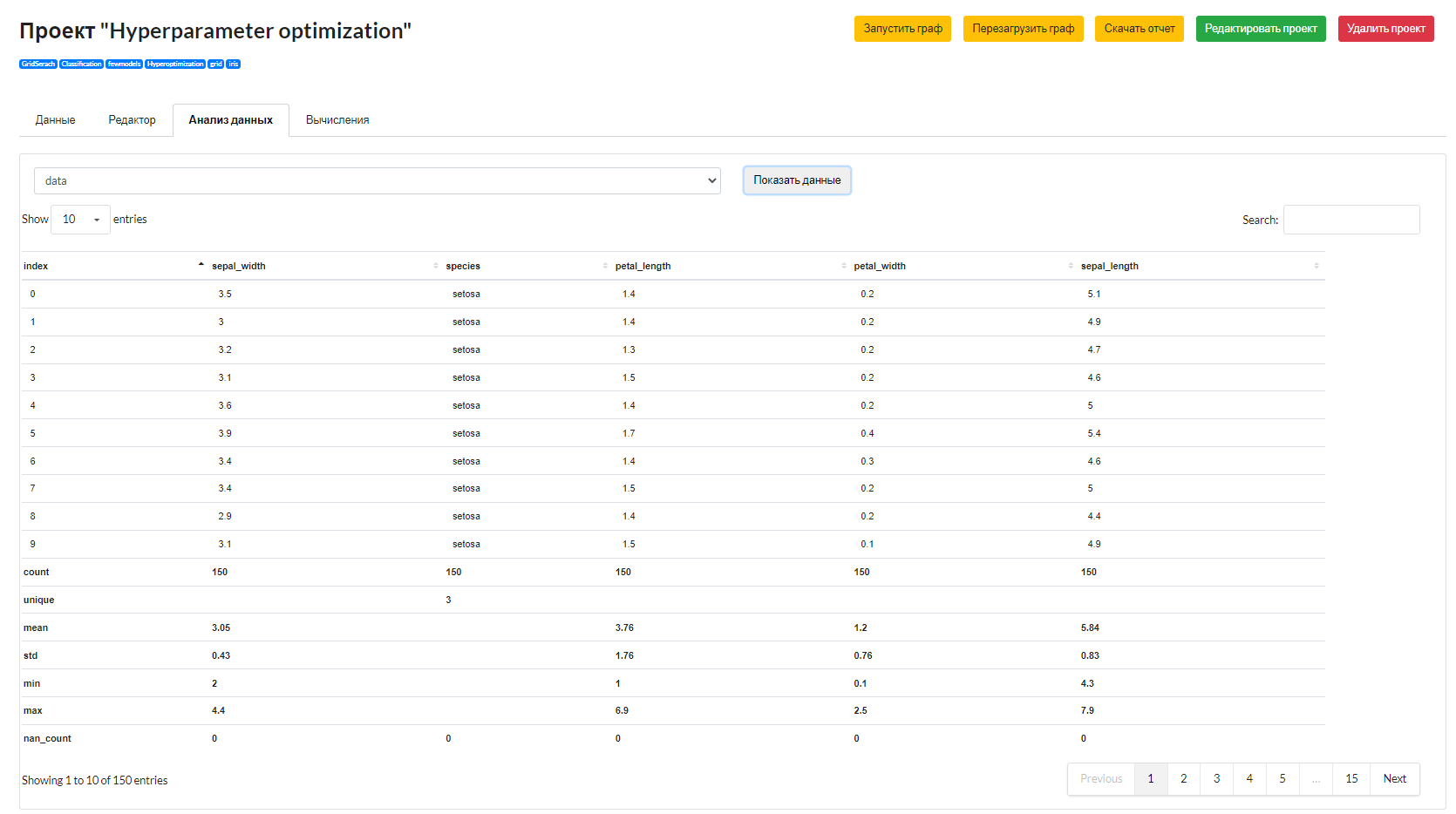


Рисунок 3.2.14 – Просмотр входных данных проекта подбора гиперпараметров моделей данных «Ирисы»

1. Процесс вычисления, осуществляемого моделью подбора параметров модели, в разделе «Вычисления» (рис. 3.2.15).

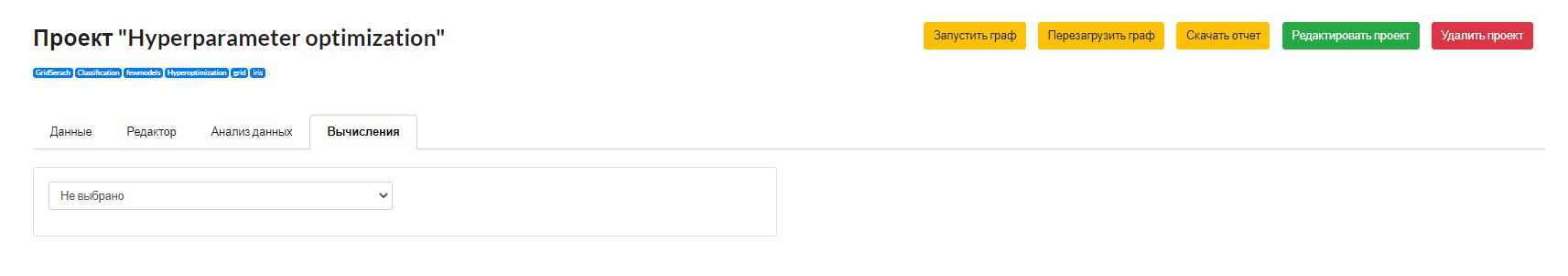


Рисунок 3.2.15 – Раздел «Вычисления» проекта подбора гиперпараметров данных «Ирисы»

* 1. Выбор узла с моделью GridSearchCV (рис. 3.2.16).

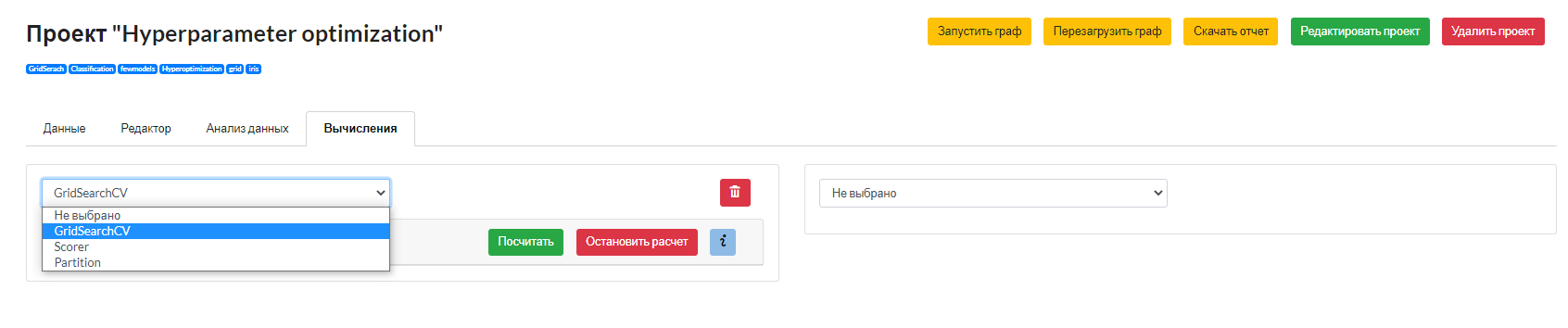


Рисунок 3.2.16 – Выбор узла с моделью GridSearchCV в разделе «Вычисления»

* 1. Настройка модели, а именно установка параметров, по которым будет осуществлен подбор параметров для данных ирисов, в меню «Расширенные настройки» (рис. 3.2.17).

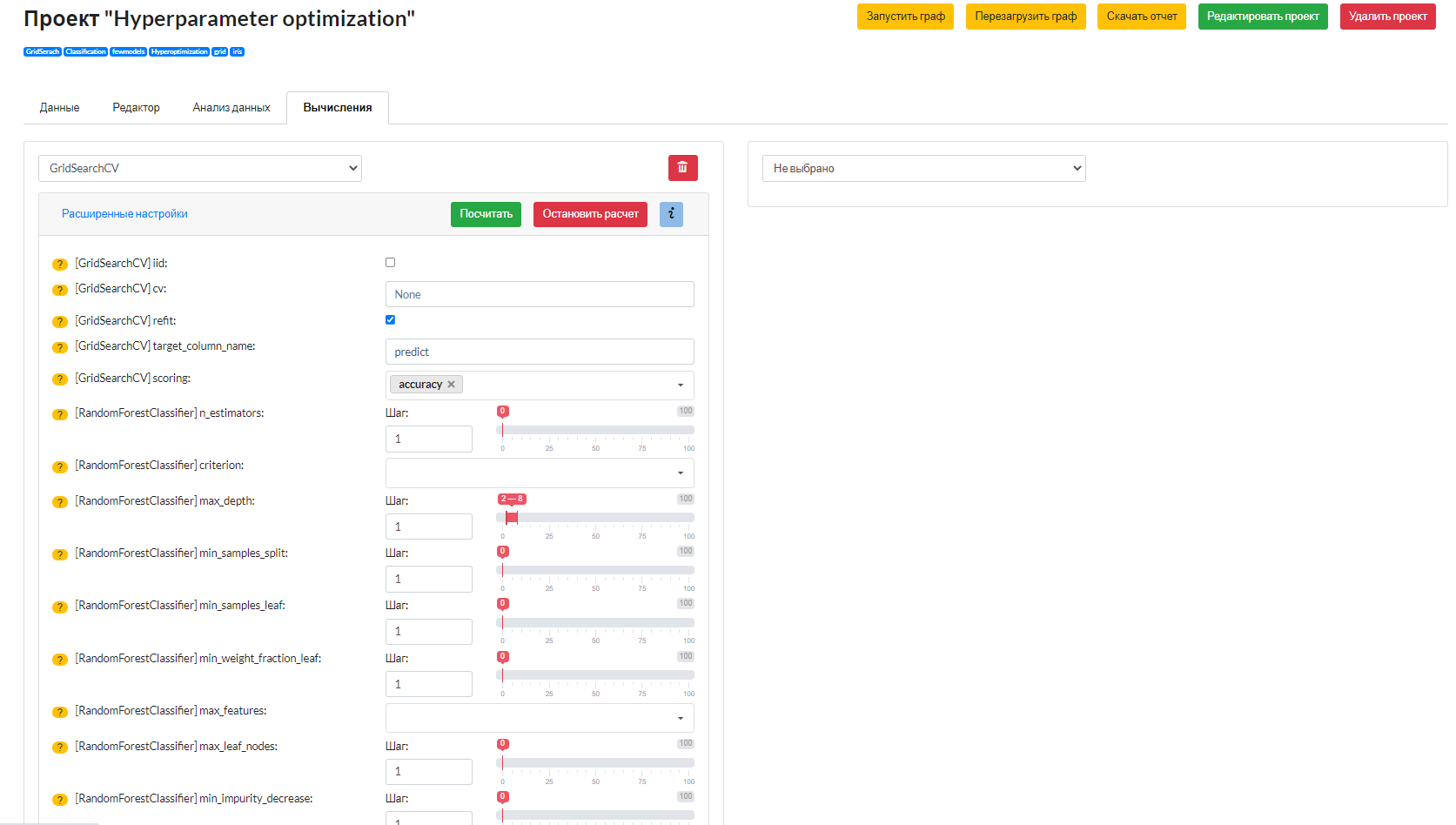


Рисунок 3.2.17 – Меню «Расширенные настройки» проекта подбора гиперпараметров моделей «Ирисы» в разделе «Вычисления»

* 1. Переход между вкладками с параметрами, полученными в результате работы метода подбора параметров модели (рис. 3.2.18).

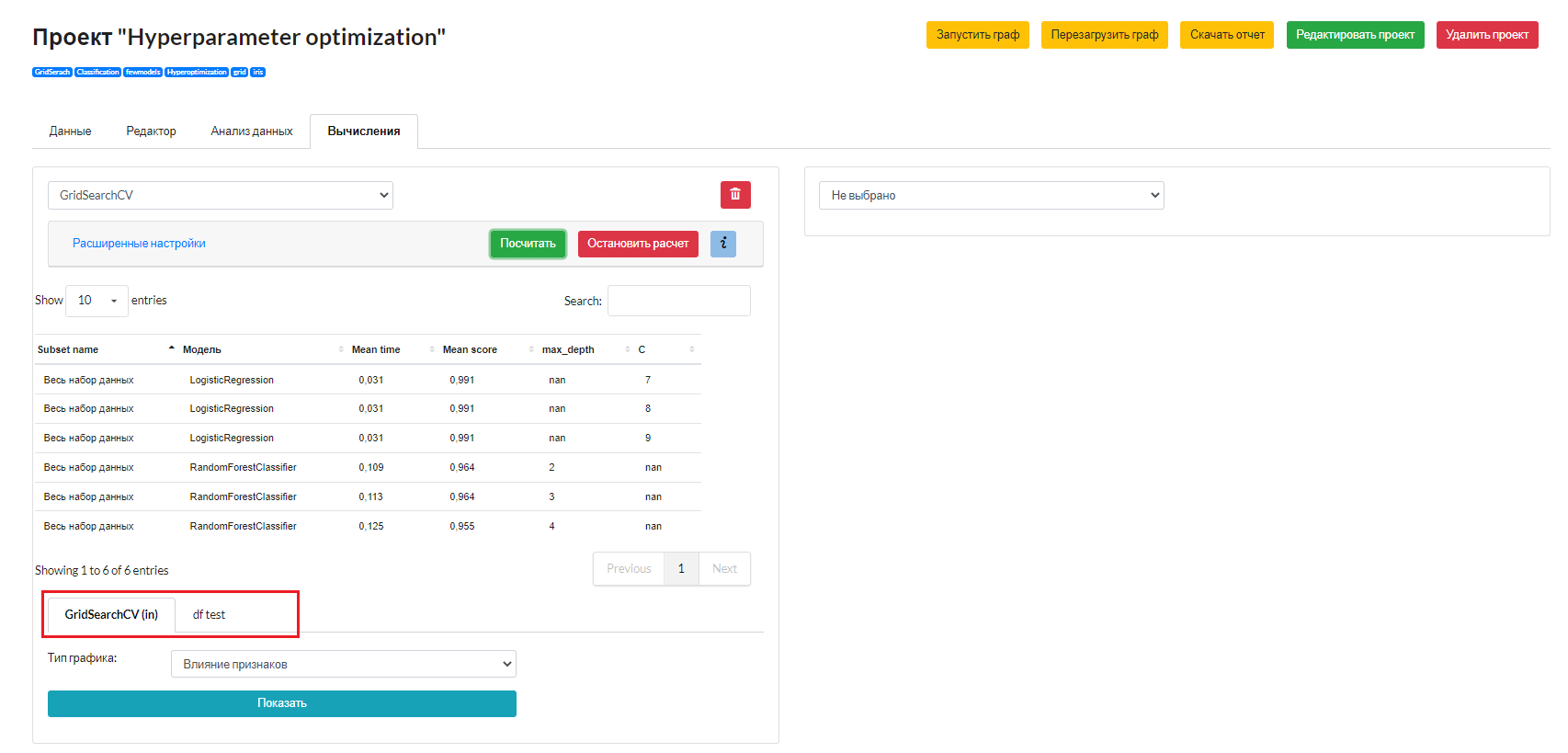


Рисунок 3.2.18 – Выбор вкладки с параметром, полученным в результате работы метода подбора параметров модели, в разделе «Вычисления»

* 1. Выбор типа визуализации результатов работы метода подбора параметров модели (рис. 3.2.19), типа данных (3.2.20) и его визуализация (рис. 3.2.21).

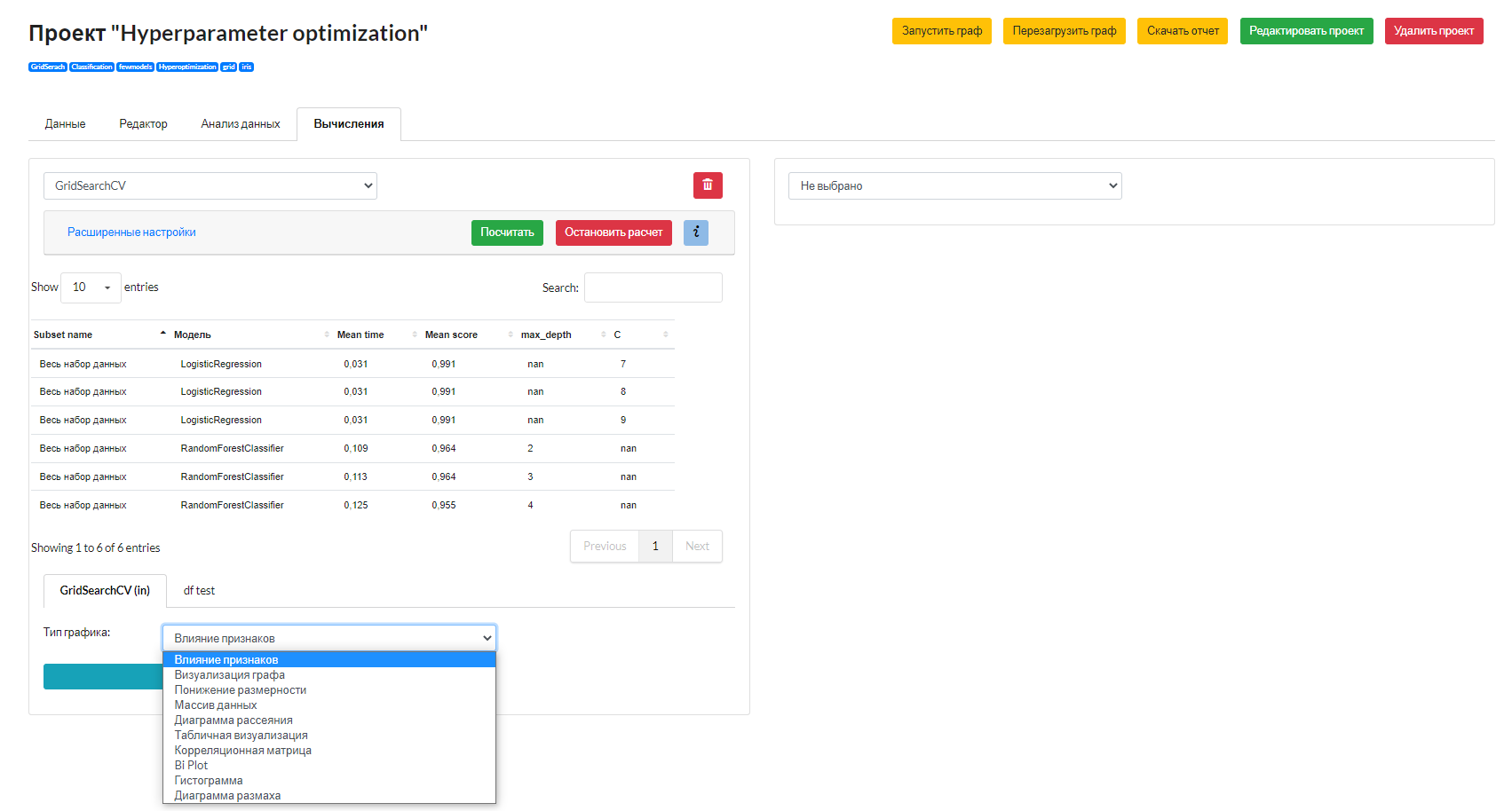


Рисунок 3.2.19 – Выбор типа визуализации результатов работы проекта подбора гиперпараметров моделей данных «Ирисы» в разделе «Вычисления»

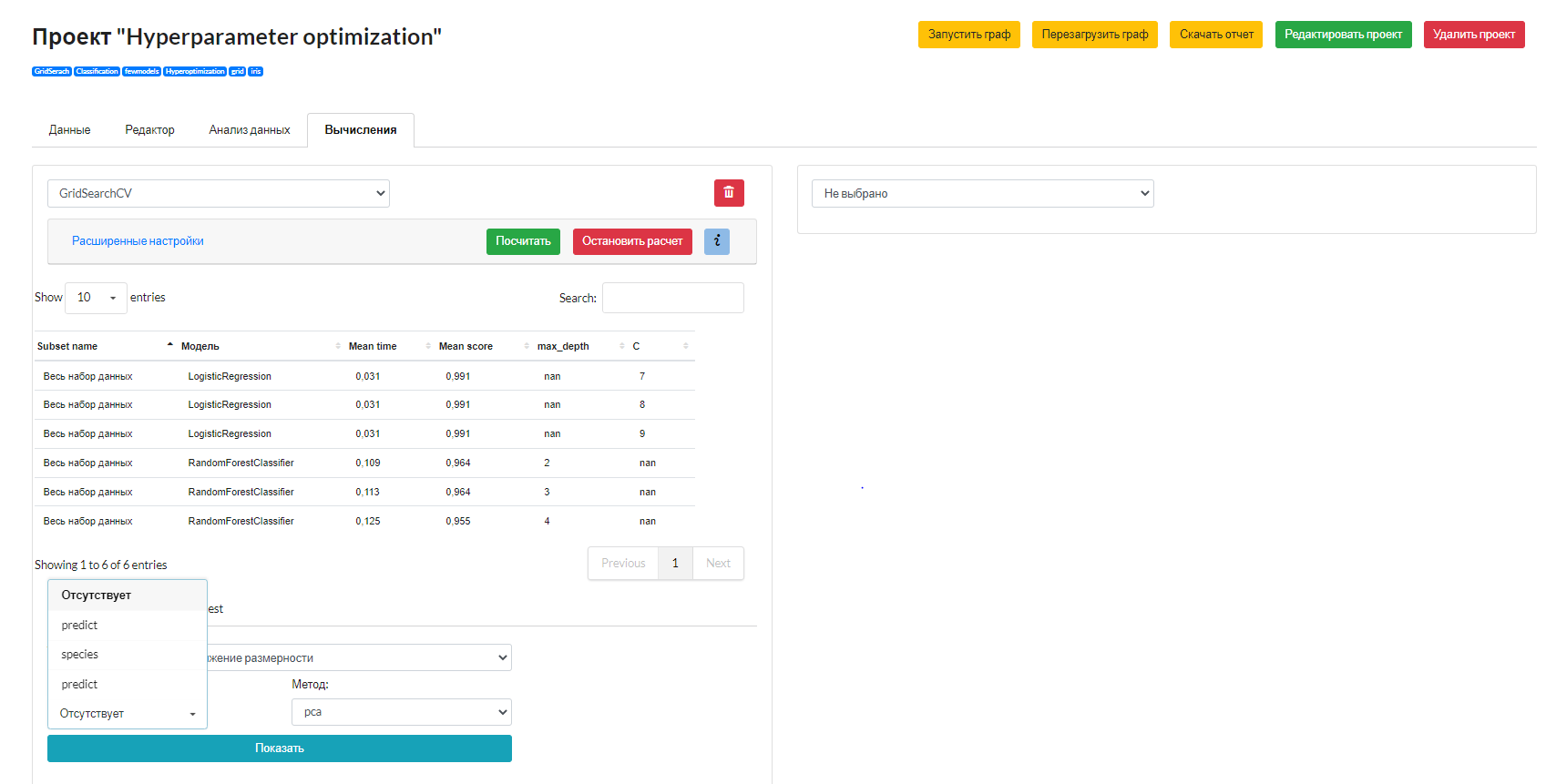


Рисунок 3.2.20 – Выбор типа данных результатов работы проекта подбора гиперпараметров моделей данных «Ирисы» в разделе «Вычисления»

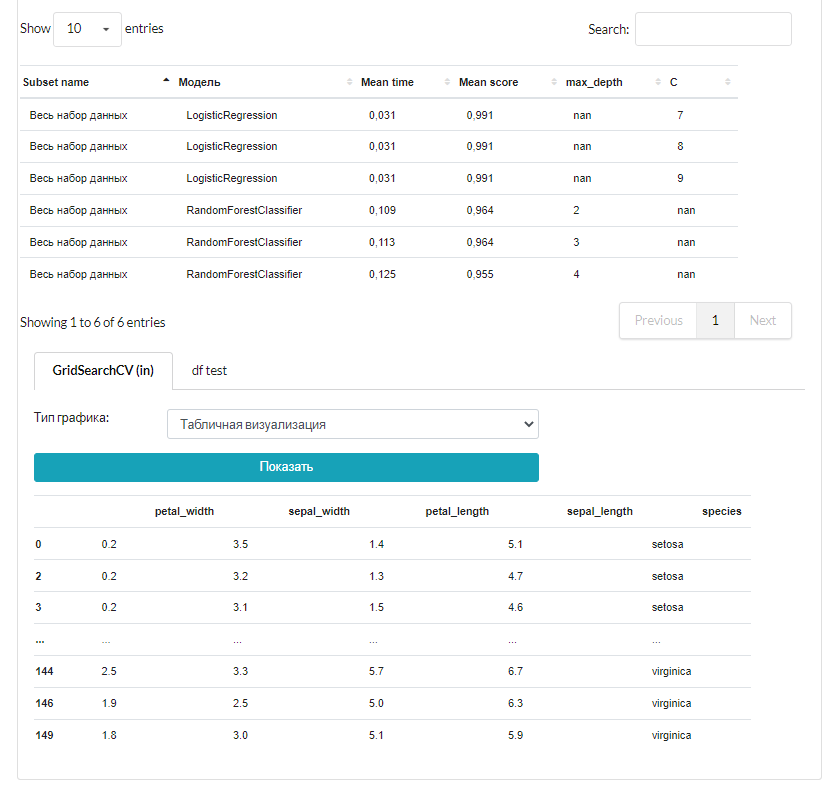


Рисунок 3.2.21 – Визуализация результатов работы проекта подбора гиперпараметров моделей данных «Ирисы» в разделе «Вычисления»

1. Сохранение результатов работы проекта.
   1. Загрузка результатов работы модели, содержащих значения подобранных параметров и результаты применения этих параметров к модели, построенной на данных «Ирисы», на ПК пользователя в разделе «Вычисления» (рис. 3.2.22).

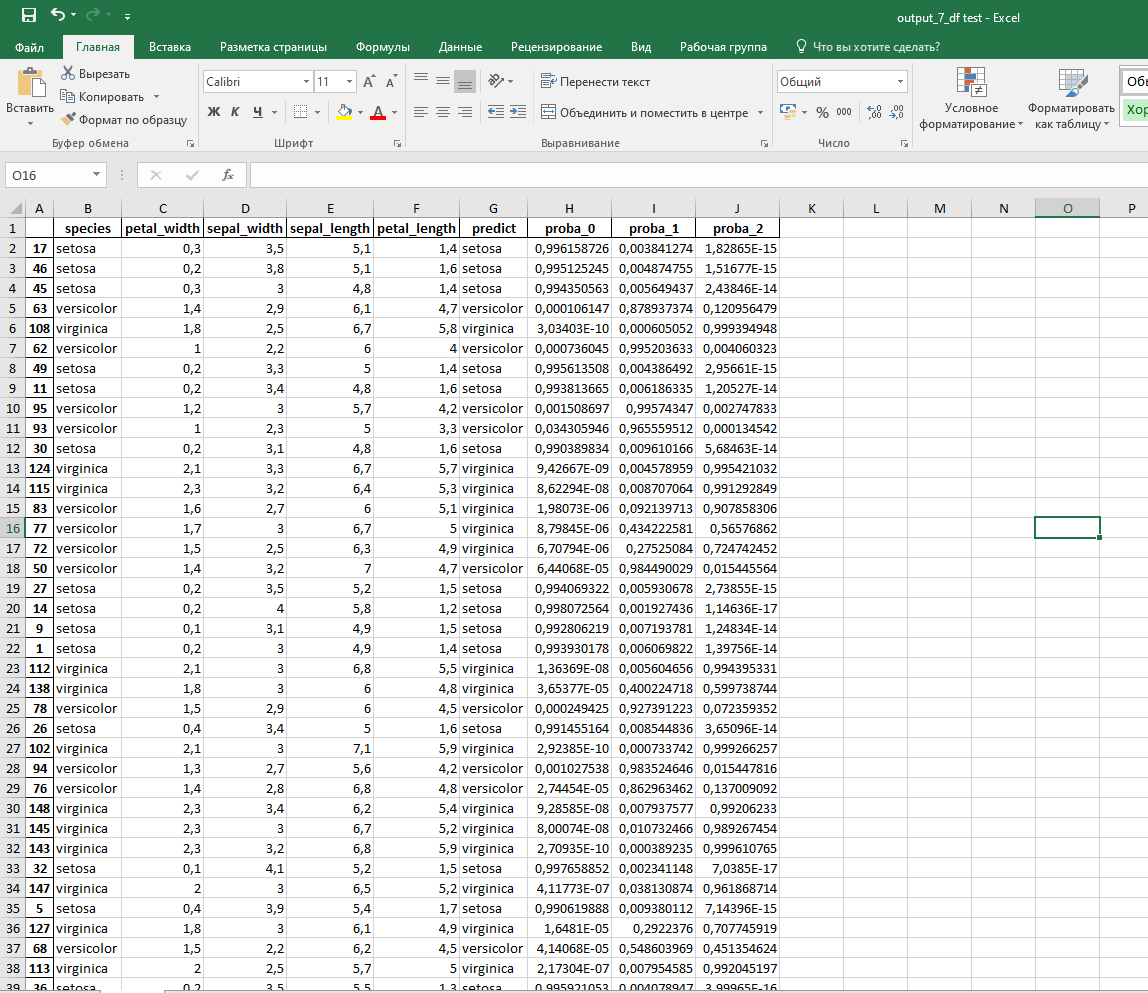


Рисунок 3.2.22 – Файл с результатами работы модели подбора гиперпараметров моделей данных «Ирисы» в разделе «Вычисления» на ПК пользователя

* 1. Загрузка отчетов работы модели на ПК пользователя из:
* Раздела «Данные»
* Раздела «Редактор»
* Раздела «Анализ данных» (рис. 3.2.23)
* Раздела «Вычисления»

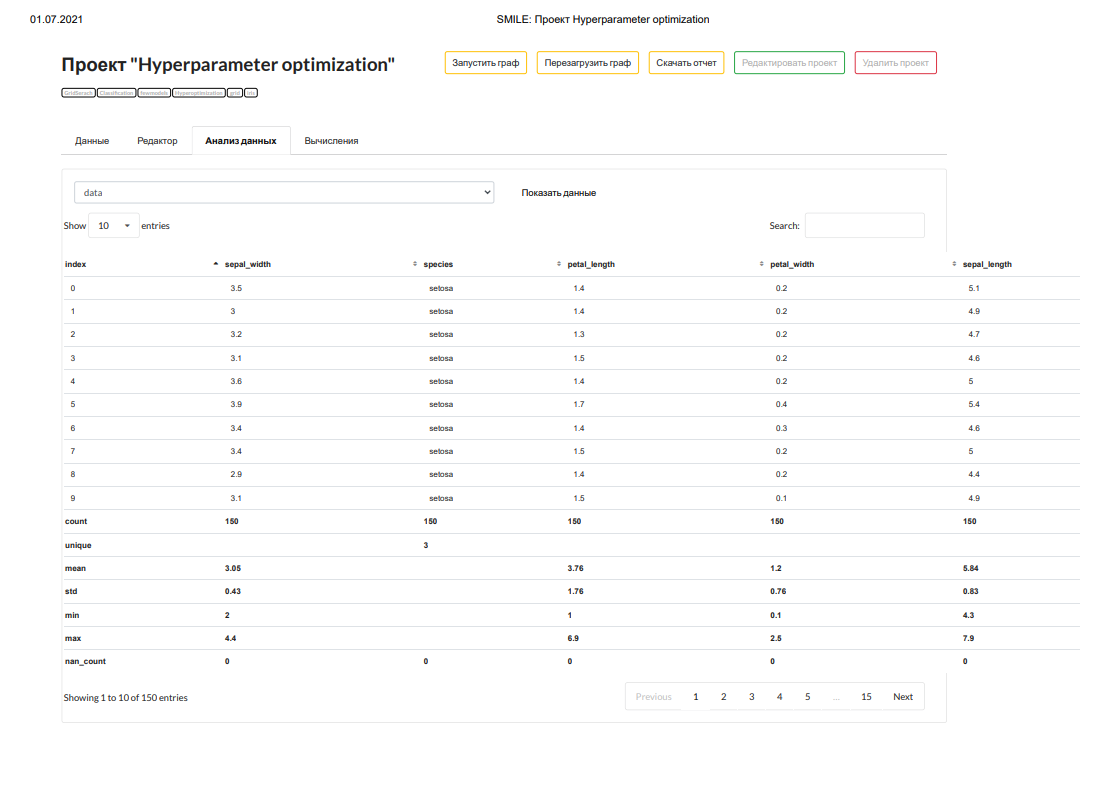


Рисунок 3.2.23 – Отчет раздела «Анализ данных» проекта подбора гиперпараметров моделей

**3.3. Ценообразование инновационных вагонов (проект «Ценообразование вагонов»)**

Работа метода линейной и гребневой регрессий на примере ценообразования инновационных вагонов включает в себя:

1. Загрузку/выбор данных о ценах типовых и инновационных вагонов в разделе «Данные» (рис. 3.3.1).

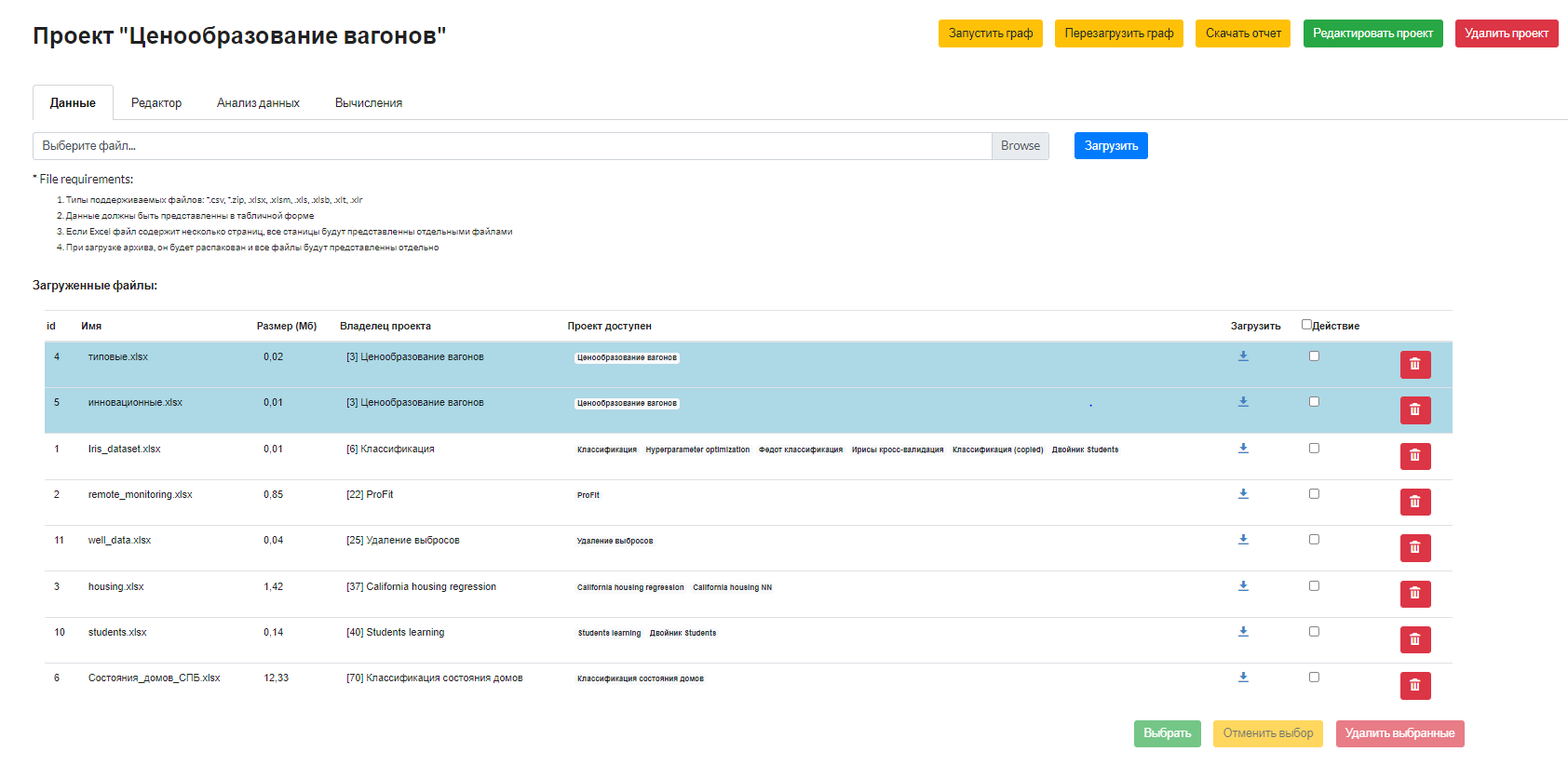


Рисунок 3.3.1 – Загрузка данных о ценах вагонов в разделе «Данные»

1. Работу с редактором графов, создаваемых для наглядного представления действий, осуществляемых с данными по вагонам, в разделе «Редактор» (рис. 3.3.2).

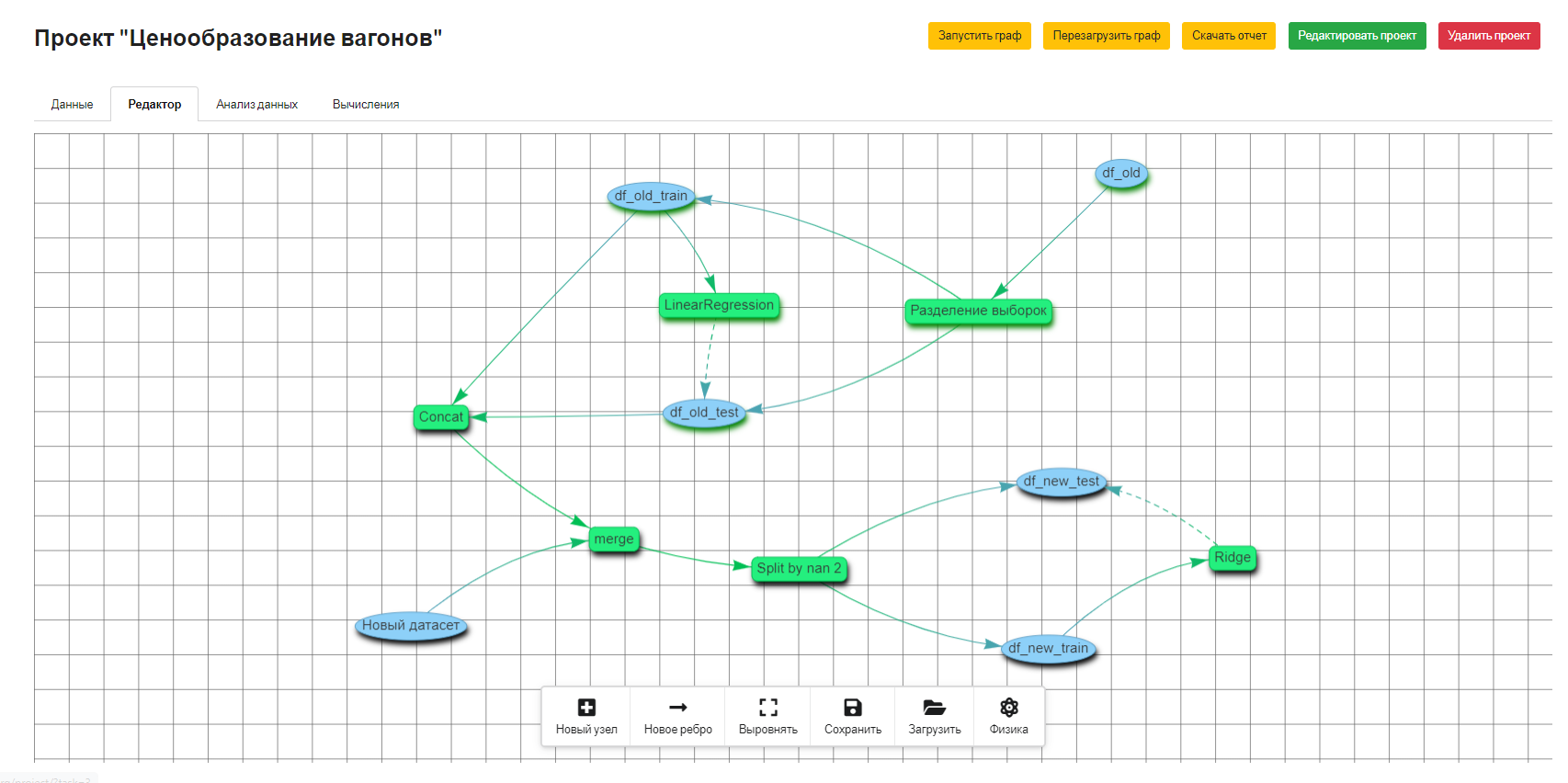


Рисунок 3.3.2 – Работа с редактором графа для данных по вагонам в разделе «Редактор»

* 1. Построение графа, а именно добавление узлов и ребер, которые используются для реализации цепи действий, осуществляемых с данными (рис. 3.3.3).

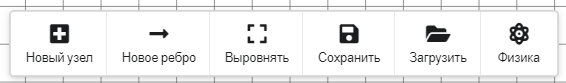


Рисунок 3.3.3 – Меню работы с графами в разделе «Редактор»

Цепь действий для задачи ценообразования инновационных вагонов:

* Чтение данных о ценах типовых вагонов (рис. 3.3.4)

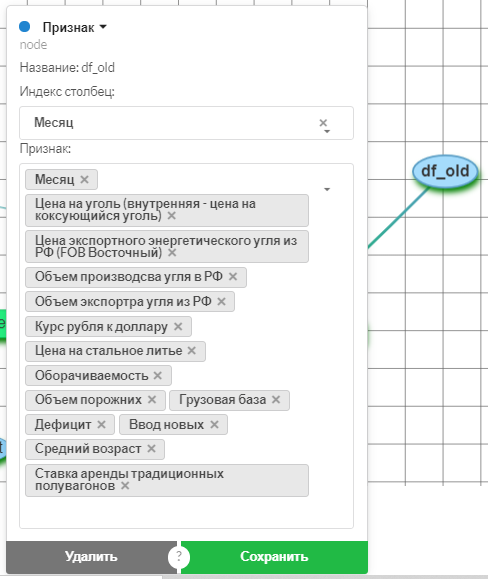


Рисунок 3.3.4 – Узел «Чтение данных о ценах типовых вагонов» графа для данных по вагонам и его свойства

* Разбиение выборки с ценами типовых вагонов (рис. 3.3.5)

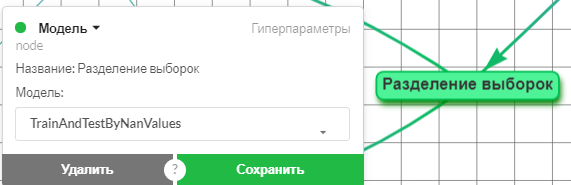


Рисунок 3.3.5 – Узел «Разбиение выборки с ценами типовых вагонов» графа для данных по вагонам и его свойства

* Тестовая выборка с ценами типовых вагонов (рис. 3.3.6)

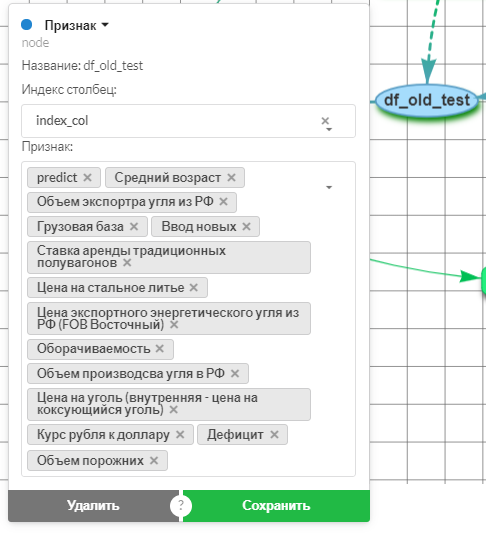


Рисунок 3.3.6 – Узел «Тестовая выборка с ценами типовых вагонов» графа для данных по вагонам и его свойства

* Тренировочная выборка с ценами типовых вагонов (рис. 3.3.7)

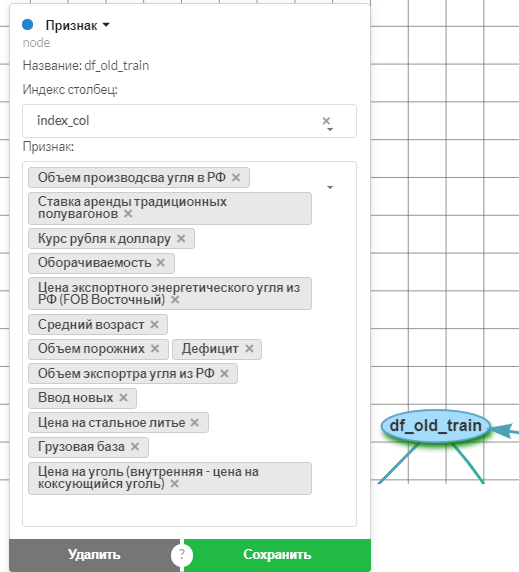


Рисунок 3.3.7 – Узел «Тренировочная выборка с ценами типовых вагонов» графа для данных по вагонам и его свойства

* Модель линейной регрессии, применяемая к данным стоимости типовых вагонов (рис.3.3.8)

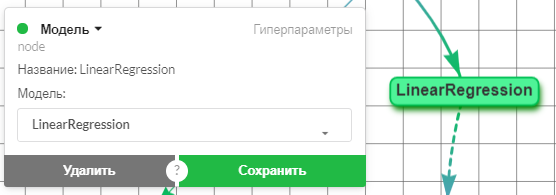


Рисунок 3.3.8 – Узел «Модель линейной регрессии, применяемая к данным стоимости типовых вагонов» графа для данных по вагонам и его свойства

* Объединение необходимых данных тестовой и тренировочной выборок цен типовых вагонов (рис. 3.3.9)

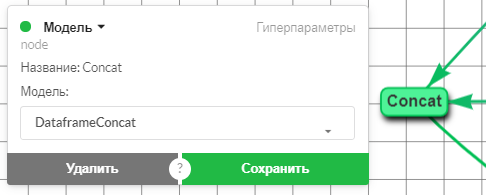


Рисунок 3.3.9 – Узел «Объединение необходимых данных тестовой и тренировочной выборок» графа для данных по вагонам и его свойства

* Чтение ранее выбранных данных тестовой и тренировочной выборок цен типовых вагонов (рис. 3.3.10)

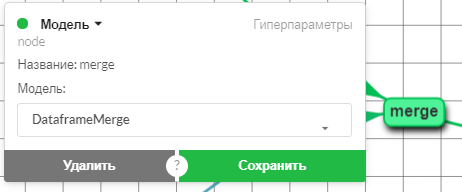


Рисунок 3.3.10 – Узел «Чтение ранее выбранных данных тестовой и тренировочной выборок» графа для данных по вагонам и его свойства

* Чтение данных о ценах инновационных вагонов (рис. 3.3.11)

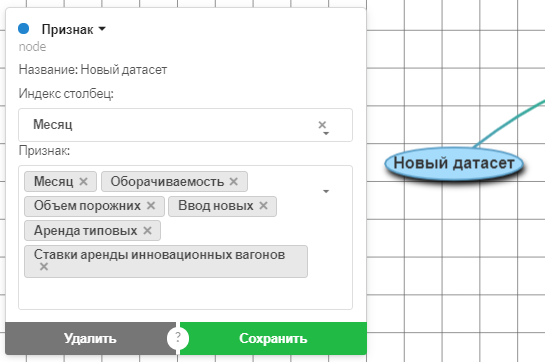


Рисунок 3.3.11 – Узел «Чтение данных о ценах инновационных вагонов» графа для данных по вагонам и его свойства

* Разбиение объединенных данных о ценах инновационных вагонов и ранее выбранных данных тестовой и тренировочной выборок цен типовых вагонов (рис. 3.3.12)

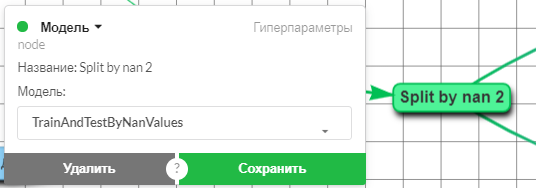


Рисунок 3.3.12 – Узел «Разбиение объединенных данных о ценах вагонов» графа для данных по вагонам и его свойства

* Тестовая выборка с ценами типовых и инновационных вагонов (рис. 3.3.13)

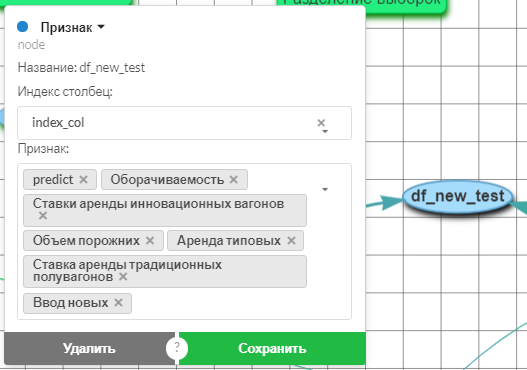


Рисунок 3.3.13 – Узел «Тестовая выборка с ценами типовых и инновационных вагонов» графа для данных по вагонам и его свойства

* Тренировочная выборка с ценами типовых и инновационных вагонов (рис. 3.3.14)

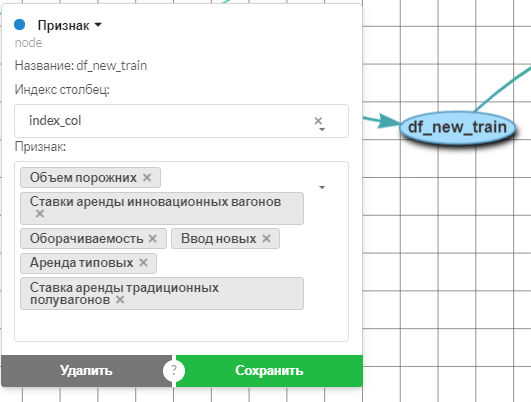


Рисунок 3.3.14 – Узел «Тренировочная выборка с ценами типовых и инновационных вагонов» графа для данных по вагонам и его свойства

* Модель гребневой регрессии, применяемая к данным стоимости типовых и инновационных вагонов (рис. 3.3.15)

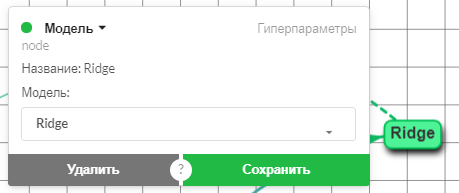


Рисунок 3.3.15 – Узел «Модель гребневой регрессии, применяемая к данным стоимости типовых и инновационных вагонов» графа для данных по вагонам и его свойства

* 1. Выбор свойств узлов/ребер (для признаков - синие узлы, а для моделей - зеленые узлы). Признаками в примере являются «Чтение данных о ценах типовых вагонов», «Тестовая выборка с ценами типовых вагонов», «Тренировочная выборка с ценами типовых вагонов», «Чтение объединенных данных тестовой и тренировочной выборок с ценами типовых вагонов», «Чтение данных о ценах инновационных вагонов», «Тестовая выборка с ценами типовых и инновационных вагонов» и «Тренировочная выборка с ценами типовых и инновационных вагонов», а моделями – «Разбиение выборки с ценами типовых вагонов», «Модель линейной регрессии, применяемая к данным стоимости типовых вагонов», «Объединение и выбор необходимых данных тестовой и тренировочной выборок цен типовых вагонов», «Объединение данных о ценах инновационных вагонов и ранее выбранных данных тестовой и тренировочной выборок цен типовых вагонов», «Разбиение объединенной выборки с ценами типовых и инновационных вагонов» и «Модель гребневой регрессии, применяемая к данным стоимости типовых и инновационных вагонов» (рис. 3.3.16).

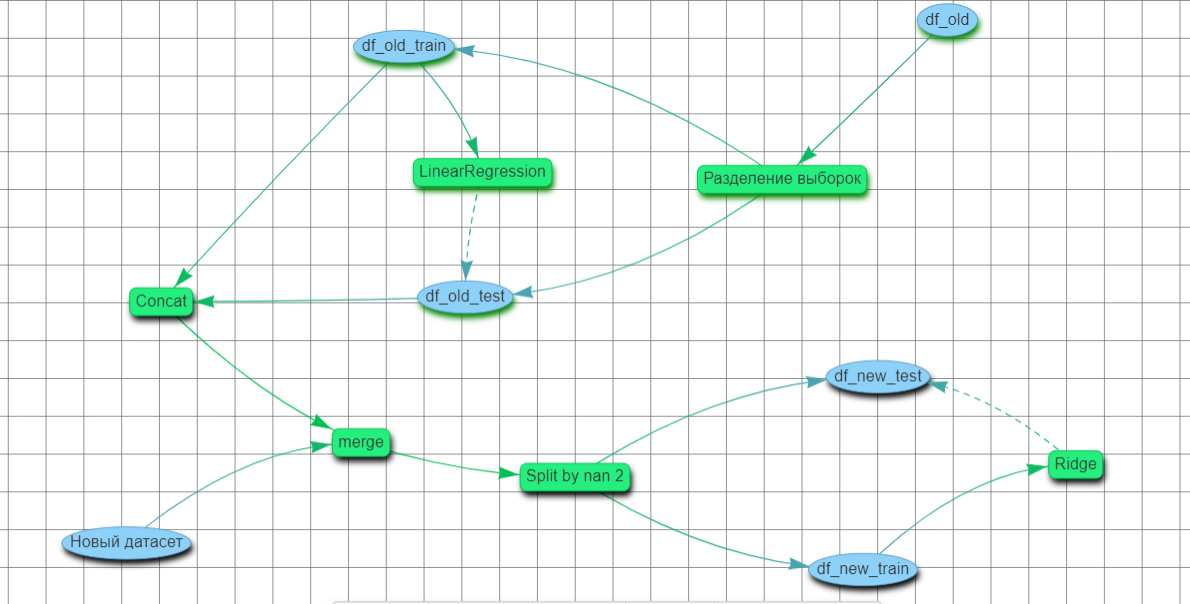


Рисунок 3.3.16 – Граф для данных по вагонам в разделе «Редактор»

1. Процесс просмотра входных данных проекта ценообразования инновационных вагонов, в разделе «Анализ данных» (рис. 3.3.17-3.3.18).

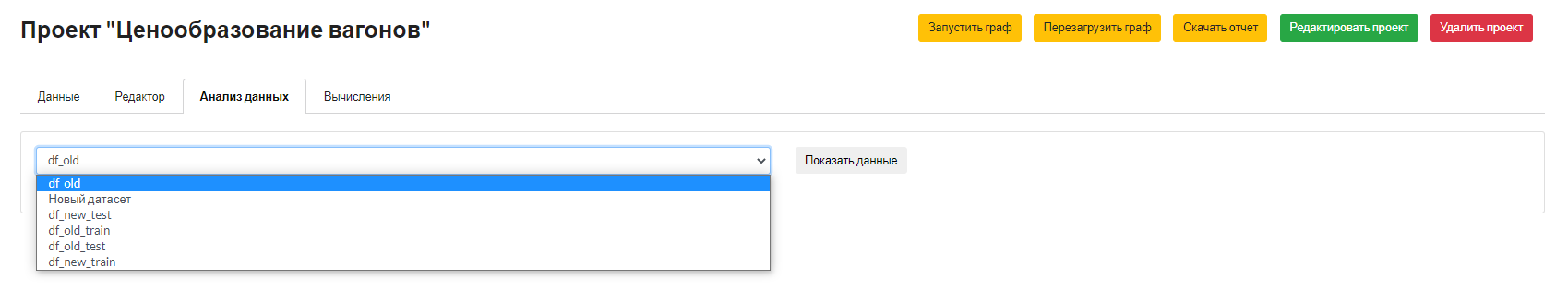


Рисунок 3.3.17 – Раздел «Анализ данных» проекта ценообразования инновационных вагонов

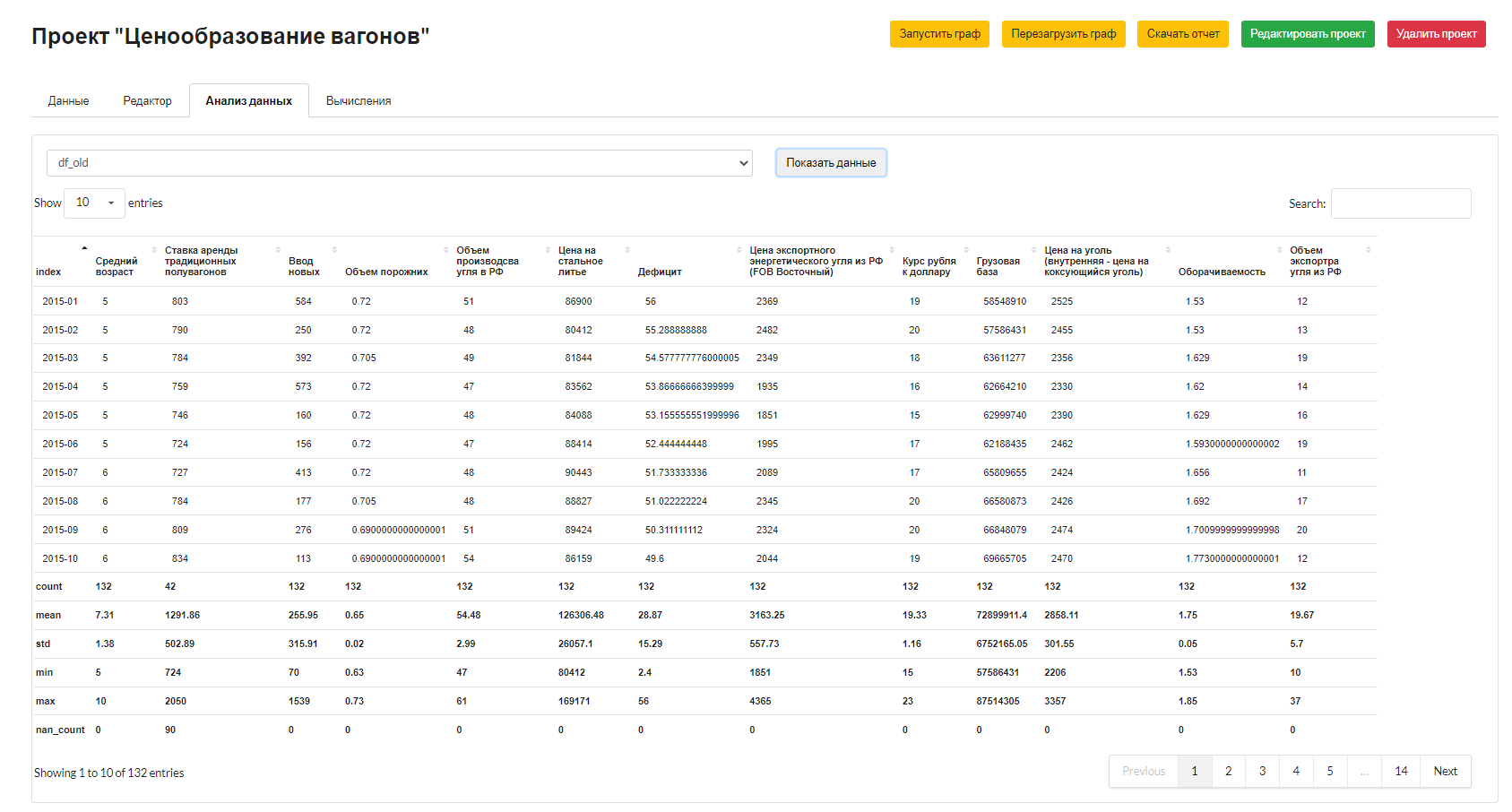


Рисунок 3.3.18 – Просмотр входных данных проекта ценообразования вагонов

1. Процесс вычисления, осуществляемого моделями линейной регрессии ценообразования типовых и гребневой регрессии ценообразования инновационных вагонов, в разделе «Вычисления» (рис. 3.3.19).

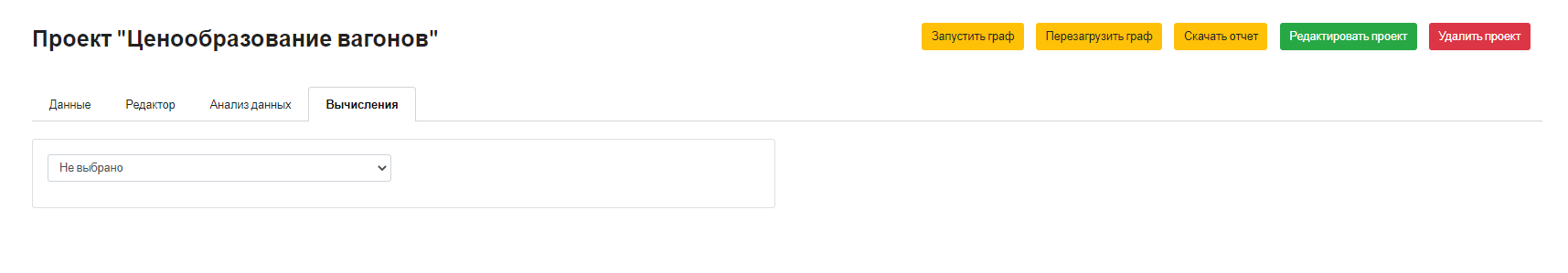


Рисунок 3.3.19 – Раздел «Вычисления» проекта ценообразования инновационных вагонов

* 1. Выбор узла с моделью гребневой регрессии для инновационных вагонов (рис. 3.3.20).

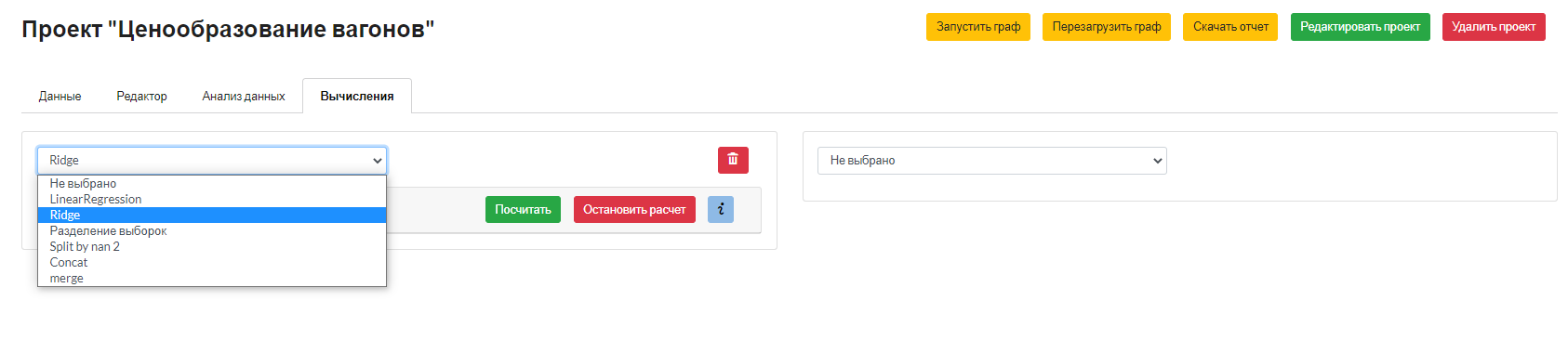


Рисунок 3.3.20 – Выбор узла с моделью гребневой регрессии для инновационных вагонов в разделе «Вычисления»

* 1. Настройка модели, а именно установка параметров, по которым будет выполнена гребневая регрессия для инновационных вагонов, в меню «Расширенные настройки» (рис. 3.3.21).

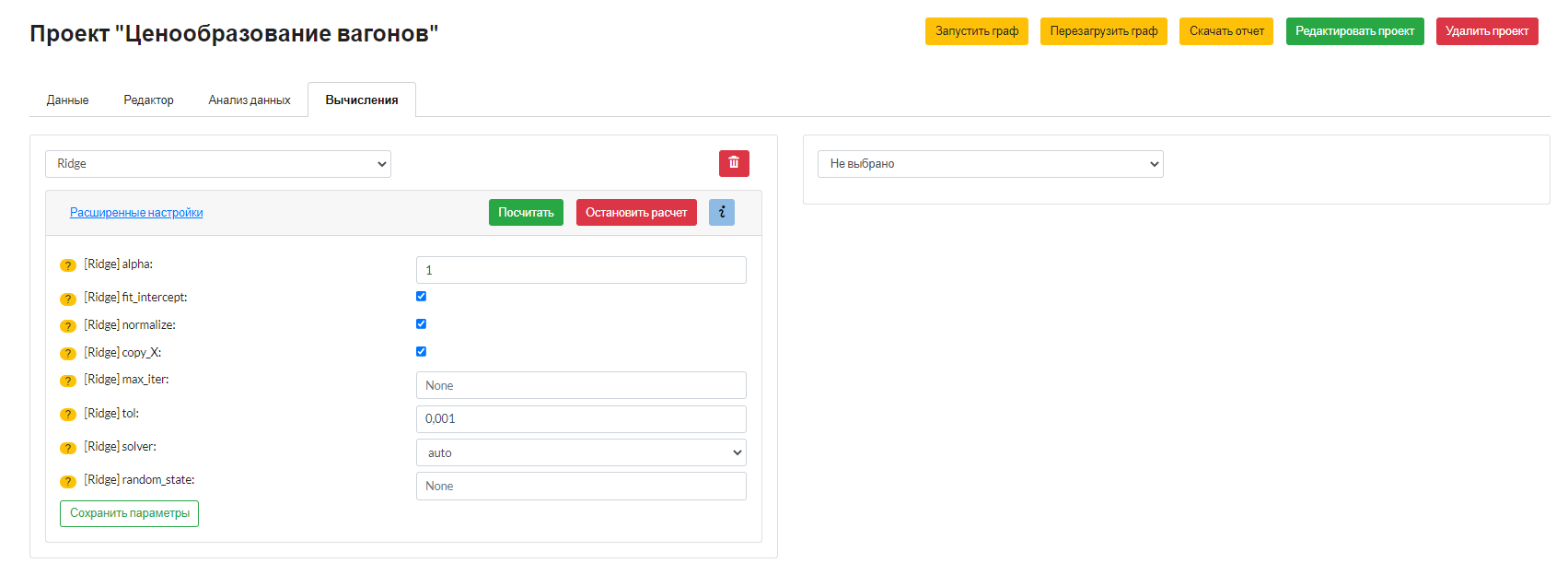


Рисунок 3.3.21 – Меню «Расширенные настройки» гребневой регрессии для инновационных вагонов в разделе «Вычисления»

* 1. Переход между вкладками с параметрами, полученными в результате работы модели гребневой регрессии ценообразования инновационных вагонов (рис. 3.1.22).

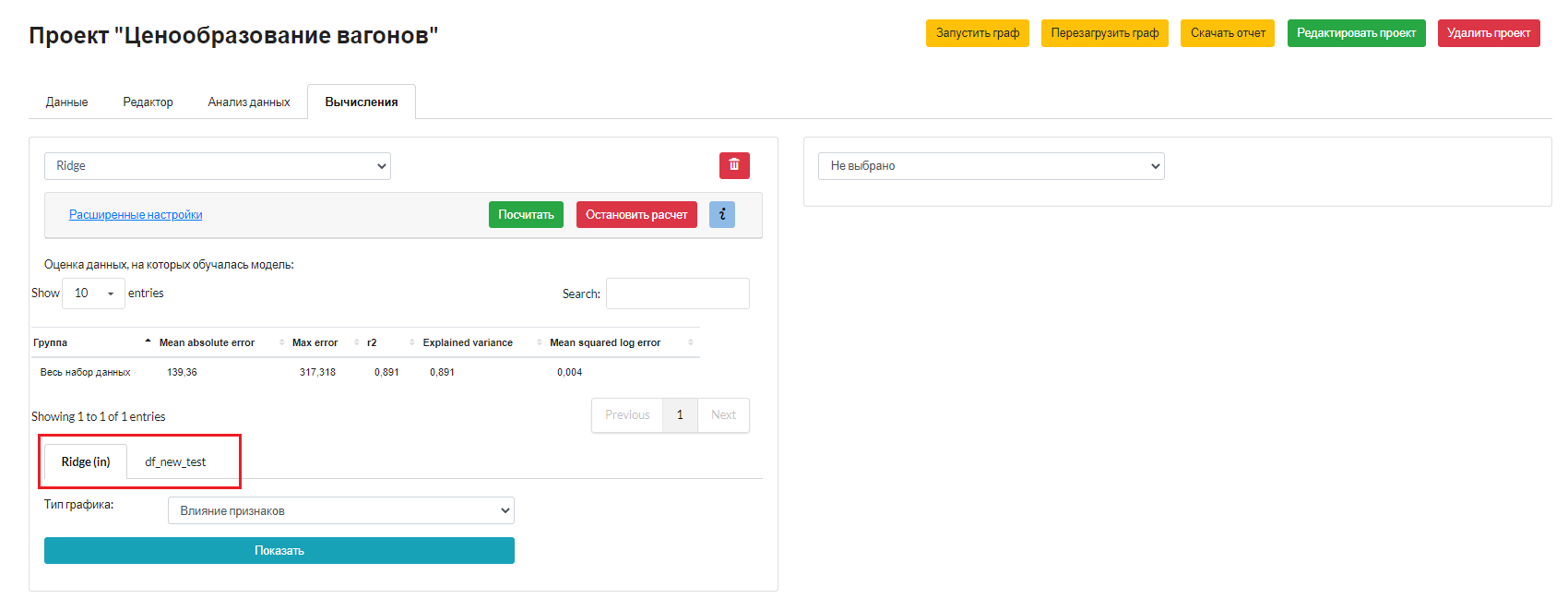


Рисунок 3.3.22 – Выбор вкладки с параметром, полученным в результате работы модели гребневой регрессии ценообразования инновационных вагонов, в разделе «Вычисления»

* 1. Выбор типа визуализации (рис. 3.3.23), типа данных (3.3.24) и визуализация результатов работы модели (рис. 3.3.25 - 3.3.26).

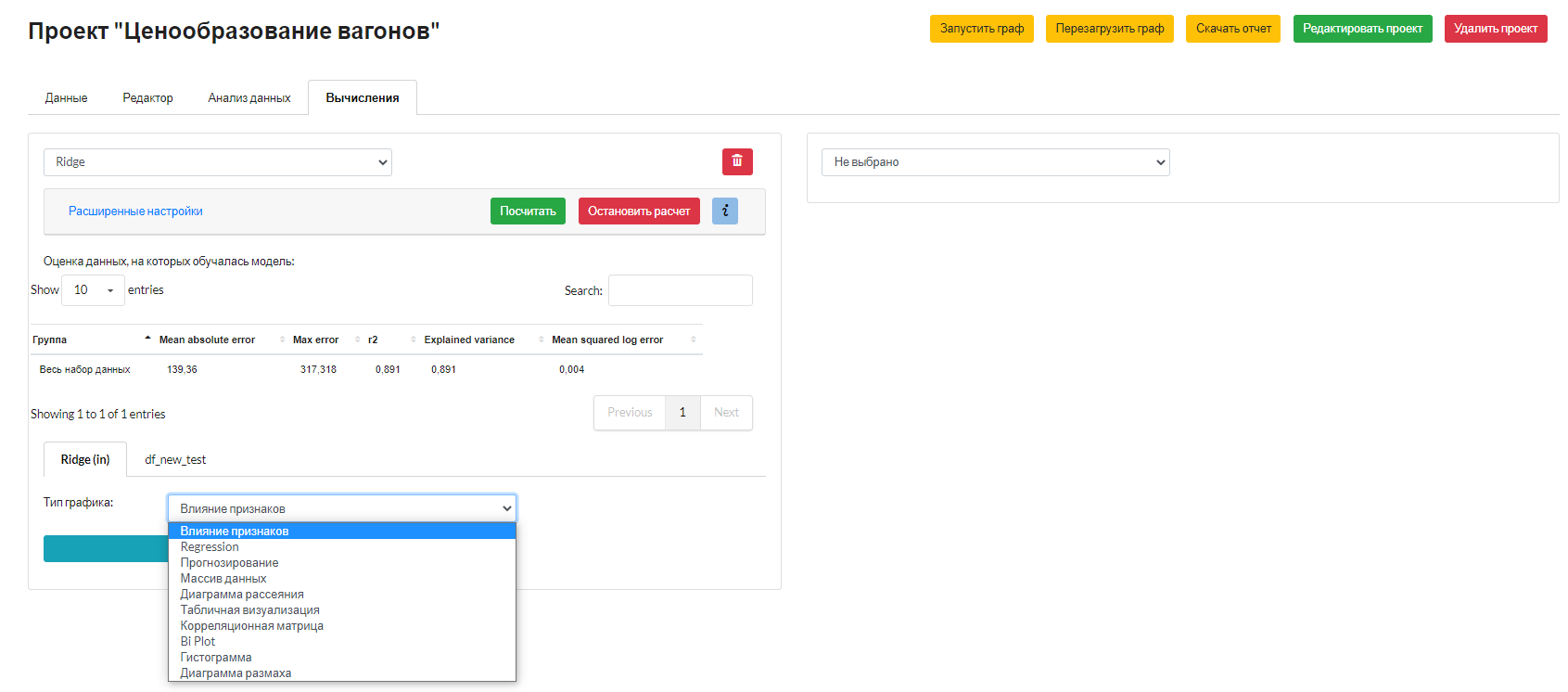


Рисунок 3.3.23 – Выбор типа визуализации результатов работы модели гребневой регрессии по данным цен инновационных вагонов в разделе «Вычисления»

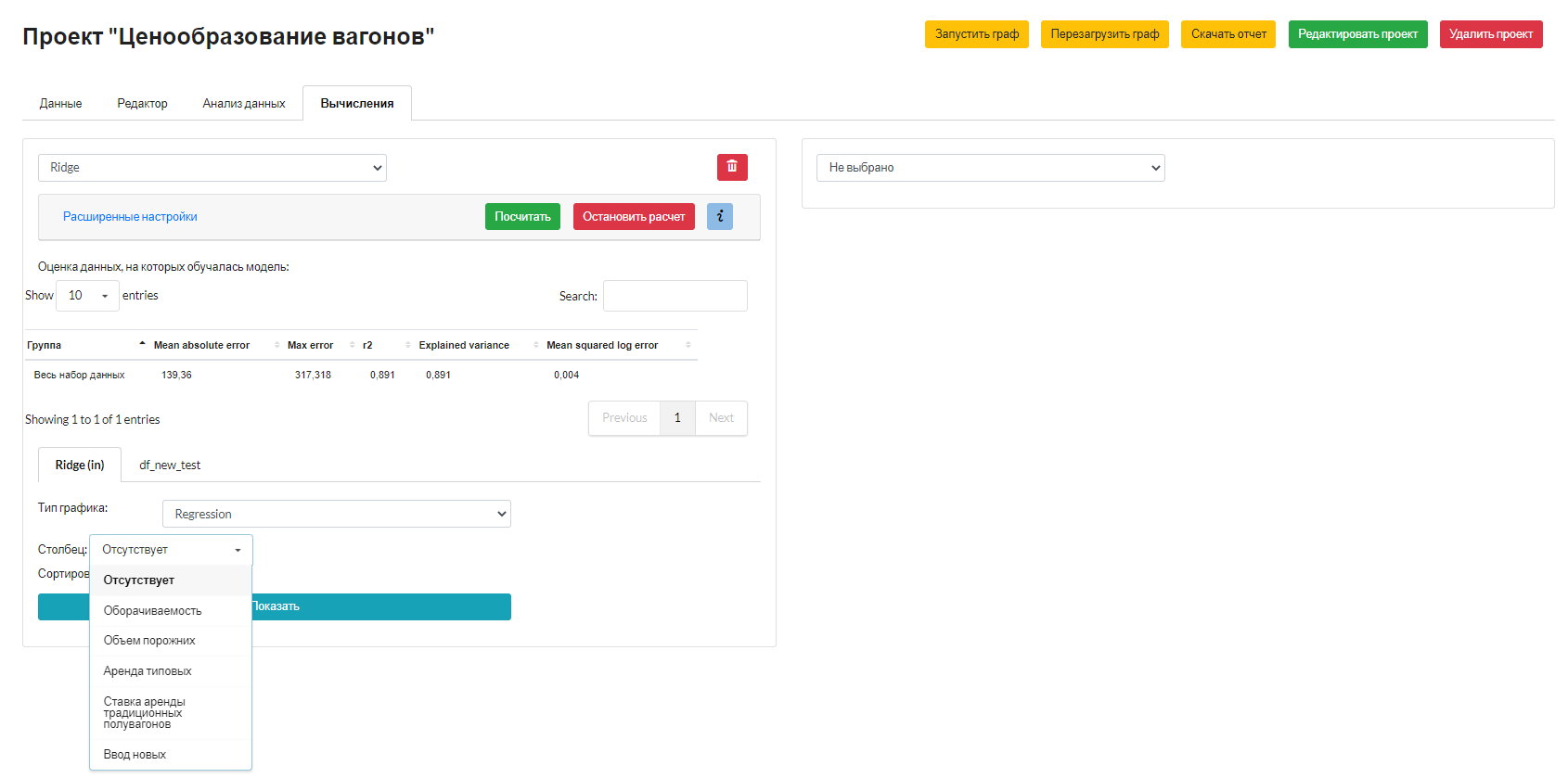


Рисунок 3.3.24 – Выбор типа данных результатов работы модели гребневой регрессии по данным цен инновационных вагонов в разделе «Вычисления»

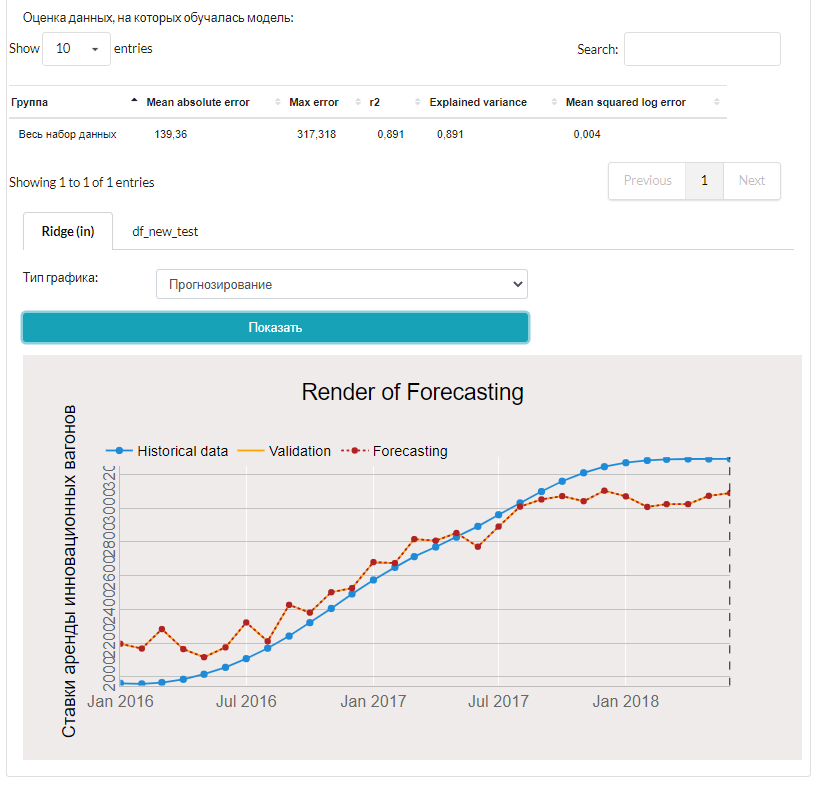


Рисунок 3.3.25 – Визуализация результатов работы модели гребневой регрессии по данным цен инновационных вагонов в разделе «Вычисления»

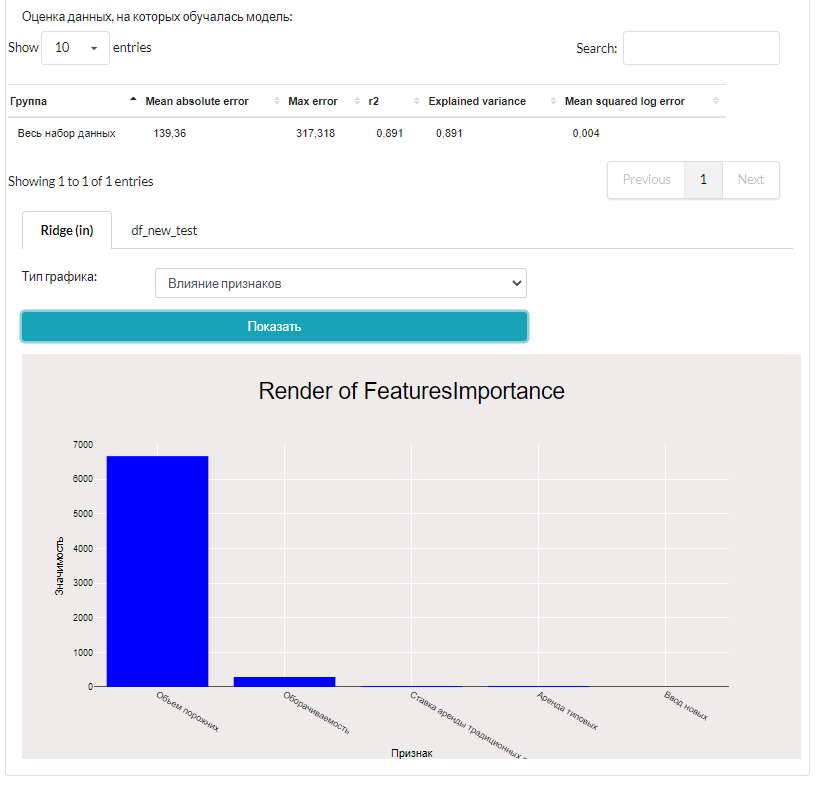


Рисунок 3.3.26 – Визуализация результатов работы модели гребневой регрессии по данным цен инновационных вагонов в разделе «Вычисления»

1. Сохранение результатов работы проекта:
   1. Загрузка результатов работы модели, содержащих предсказанные значения цен инновационных вагонов, на ПК пользователя в разделе «Вычисления» (рис. 3.3.27).

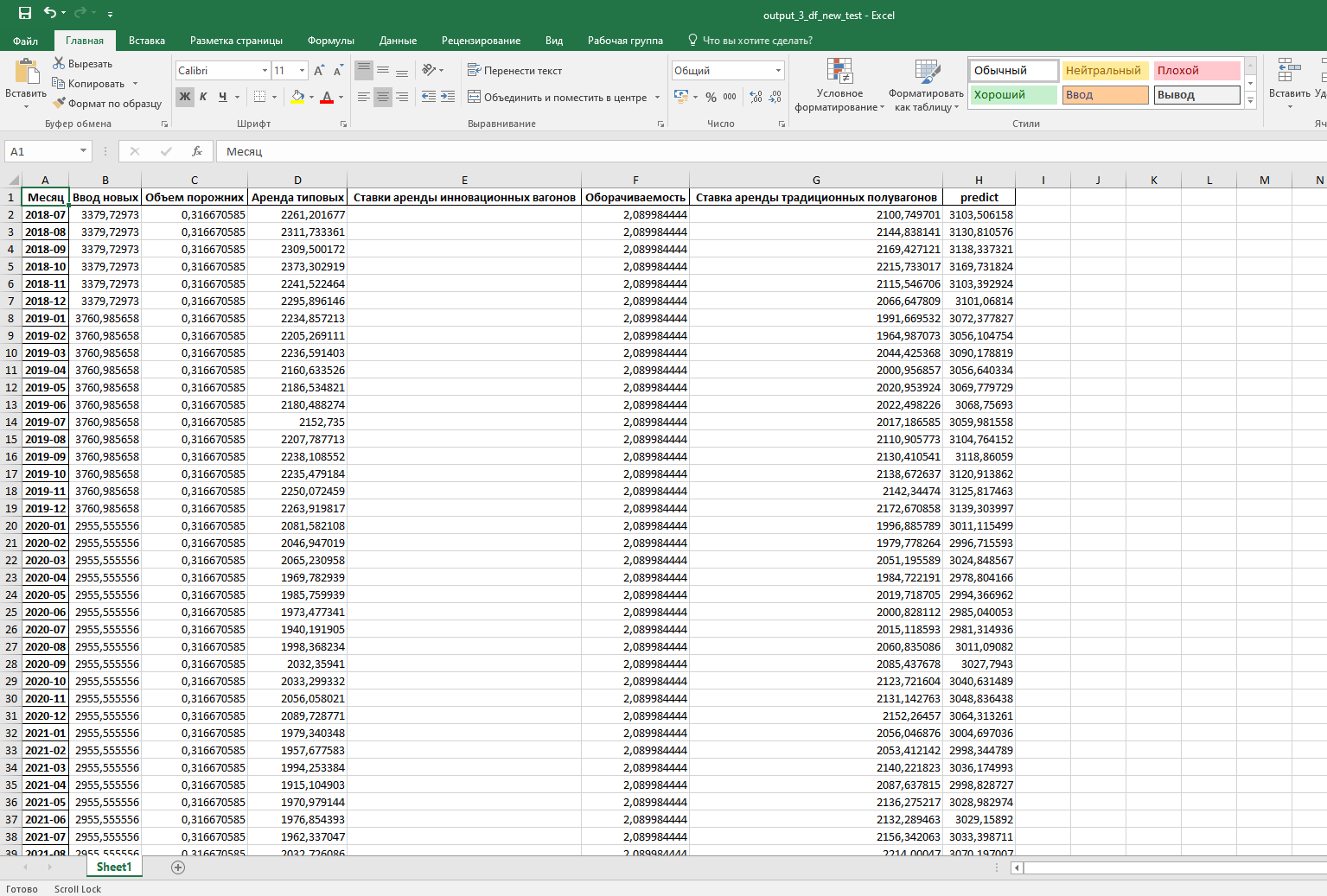


Рисунок 3.3.27 – Файл с результатами работы модели гребневой регрессии по данным цен инновационных вагонов в разделе «Вычисления» на ПК пользователя

* 1. Загрузка отчетов работы модели на ПК пользователя из:
* Раздела «Данные»
* Раздела «Редактор»
* Раздела «Анализ данных»
* Раздела «Вычисления» (рис. 3.3.28)

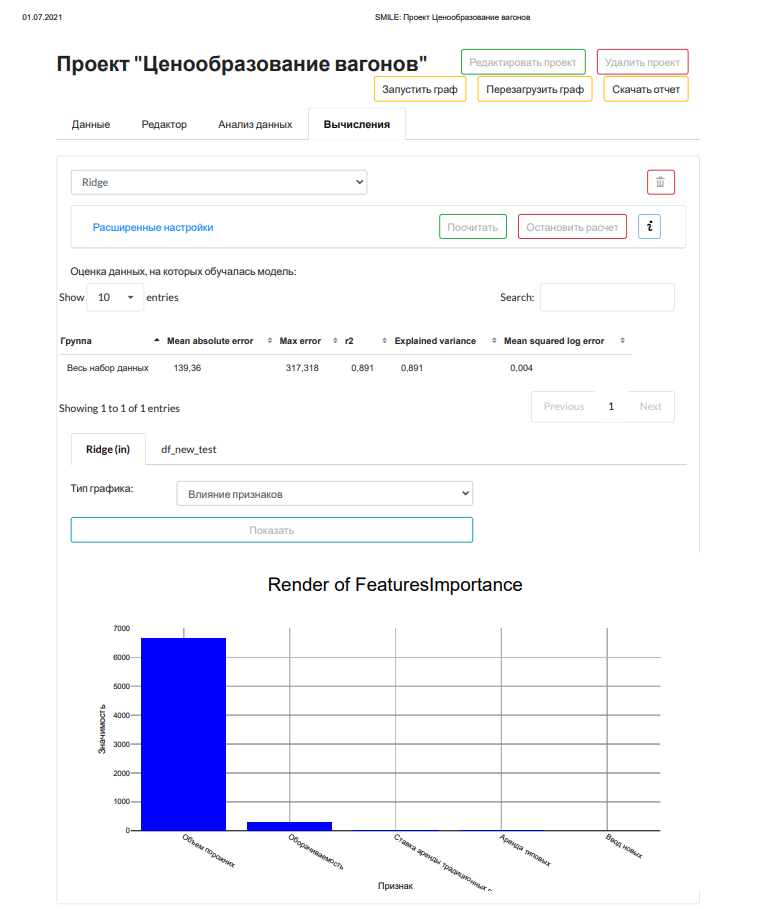


Рисунок 3.3.28 – Отчет раздела «Вычисления» проекта ценообразования инновационных вагонов

**3.4. Временной ряд (проект «Временной ряд»)**

Работа метода анализа и прогнозирования значений временного ряда на примере некоторого ряда данных, собранных в разные моменты времени, включает в себя:

1. Загрузку/выбор данных, собранных в разные моменты времени, в разделе «Данные» (рис. 3.4.1).

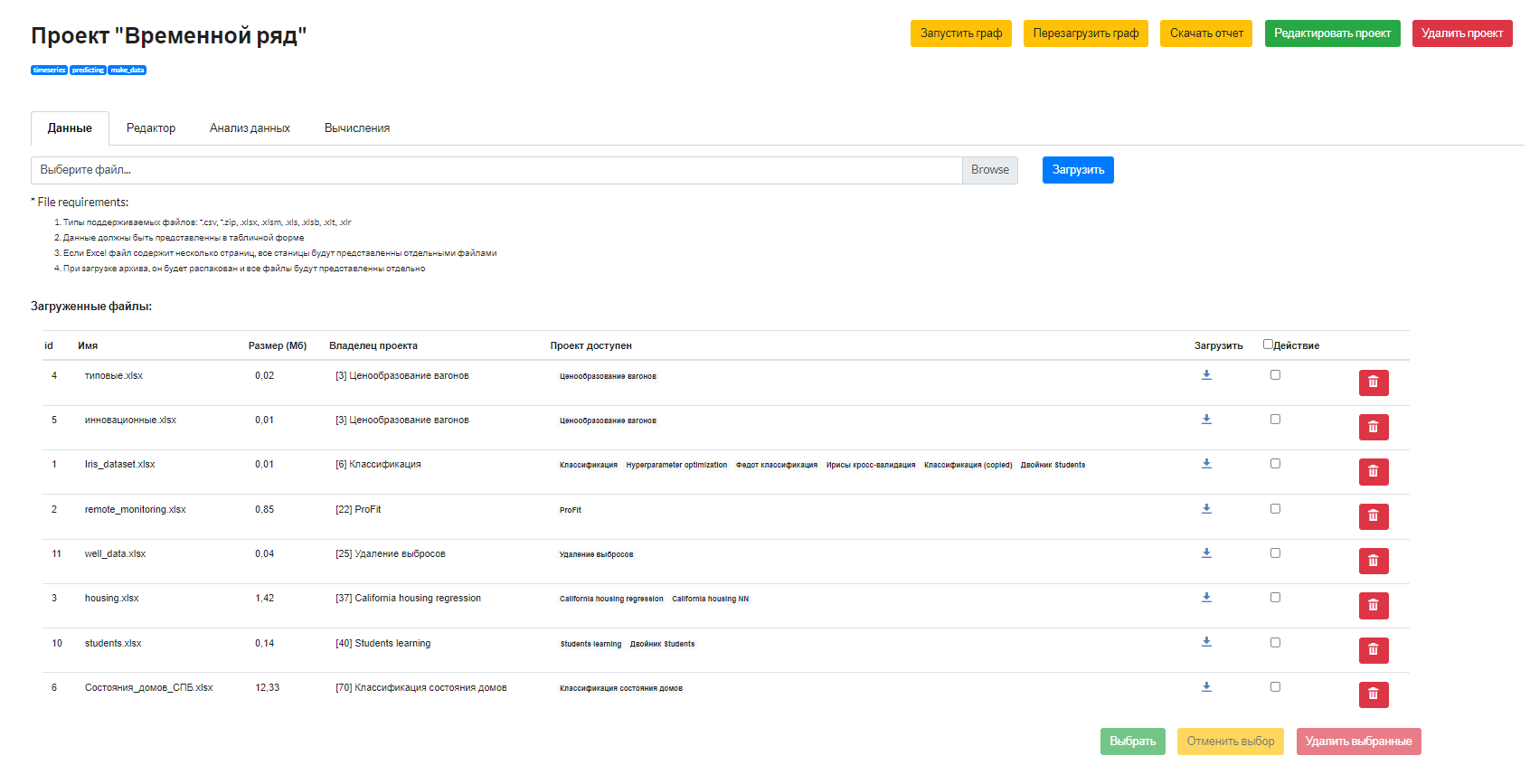


Рисунок 3.4.1 – Загрузка данных в разделе «Данные»

1. Работу с редактором графов, создаваемых для наглядного представления действий, осуществляемых с данными по вагонам, в разделе «Редактор» (рис. 3.4.2).

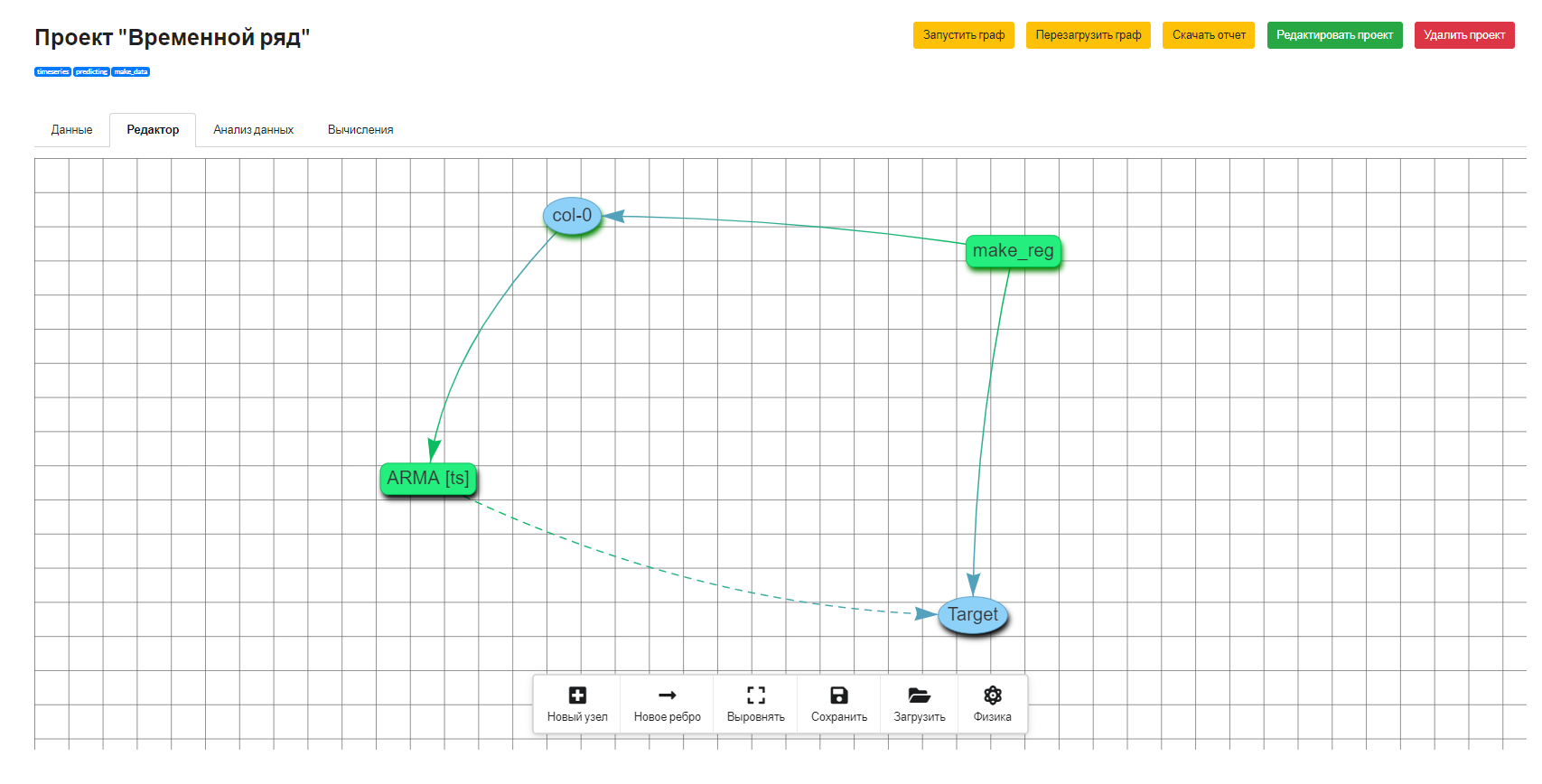


Рисунок 3.4.2 – Работа с редактором графа для данных в разделе «Редактор»

* 1. Построение графа, а именно добавление узлов и ребер, которые используются для реализации цепи действий, осуществляемых с данными (рис. 3.4.3).

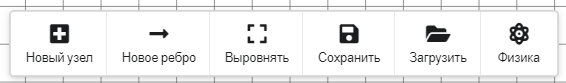


Рисунок 3.4.3 – Меню работы с графами в разделе «Редактор»

Цепь действий для задачи анализа и прогнозирования временного ряда:

* Создание ряда данных, собранных в разные моменты времени (рис. 3.4.4)

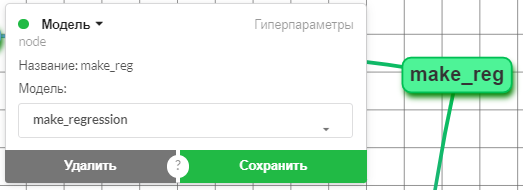


Рисунок 3.4.4 – Узел «Создание ряда данных, собранных в разные моменты времени» графа для данных по вагонам и его свойства

* Тестовая выборка данных временного ряда (рис. 3.4.5)

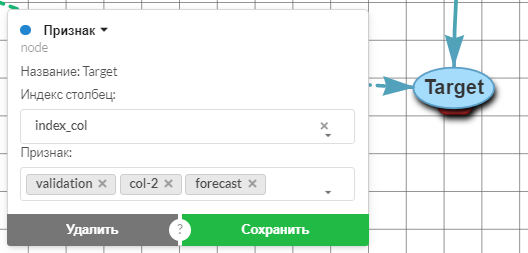


Рисунок 3.4.5 – Узел «Тестовая выборка данных временного ряда» графа для данных по вагонам и его свойства

* Тренировочная выборка данных временного ряда (рис. 3.4.6)

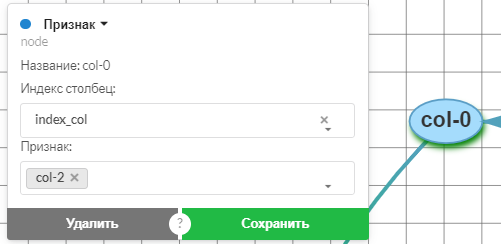


Рисунок 3.4.6 – Узел «Тренировочная выборка данных временного ряда» графа для данных по вагонам и его свойства

* Модель, предсказывающая значения временного ряда на определенное пользователем количество дней вперед (рис. 3.4.7)

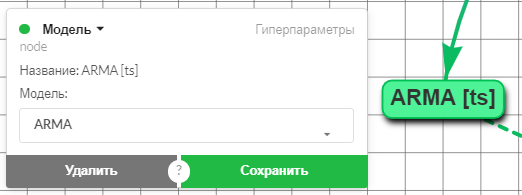


Рисунок 3.4.7 – Узел «Модель, предсказывающая значения временного ряда» графа для данных по вагонам и его свойства

* 1. Выбор свойств узлов/ребер (для признаков - синие узлы, а для моделей - зеленые узлы). Признаками в примере являются «Тренировочная выборка данных временного ряда» и «Тестовая выборка данных временного ряда», а моделями – «Создание ряда данных, собранных в разные моменты времени» и «Модель, предсказывающая значения временного ряда» (рис. 3.4.8).

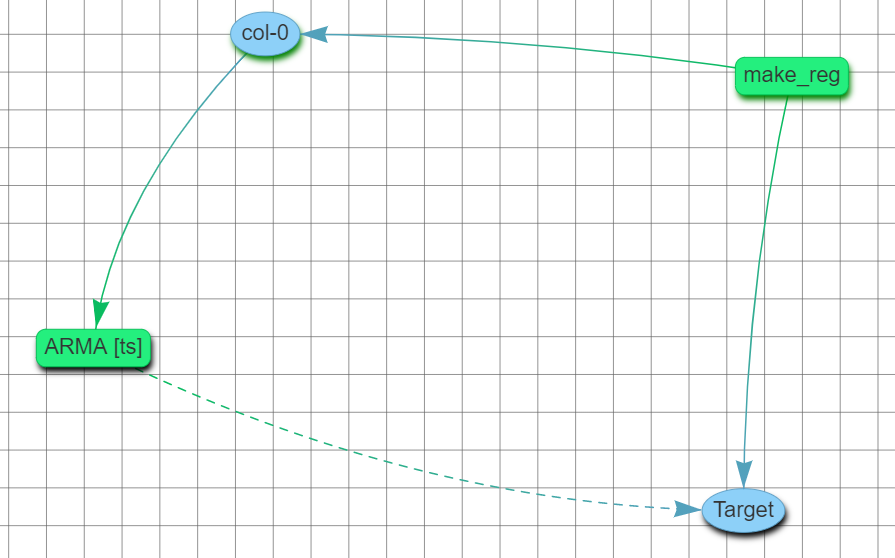


Рисунок 3.4.8 – Граф для данных временного ряда в разделе «Редактор»

1. Процесс просмотра входных данных проекта анализа и прогнозирования временного ряда, в разделе «Анализ данных» (рис. 3.4.9-3.4.10).

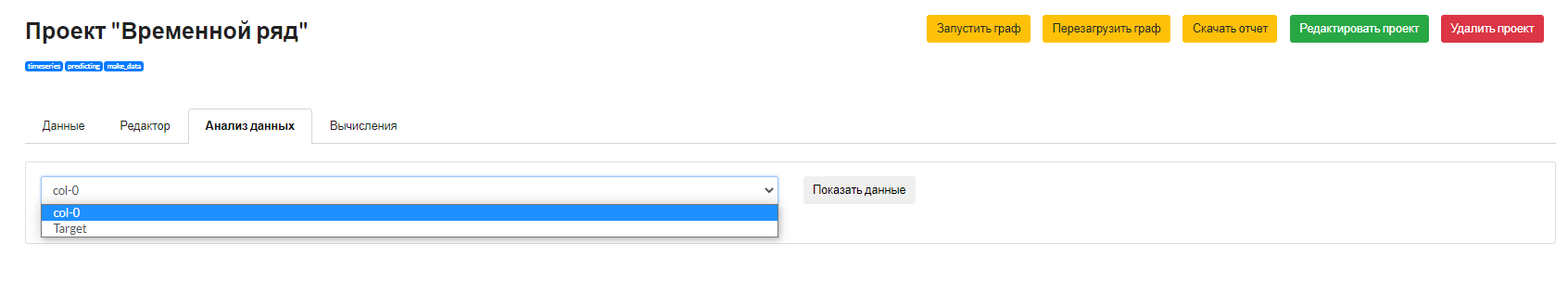


Рисунок 3.4.9 – Раздел «Анализ данных» проекта анализа и прогнозирования временного ряда



Рисунок 3.4.10 – Просмотр входных данных проекта анализа и прогнозирования временного ряда

1. Процесс вычисления, осуществляемого моделями анализа и прогнозирования временного ряда, в разделе «Вычисления» (рис. 3.4.11).

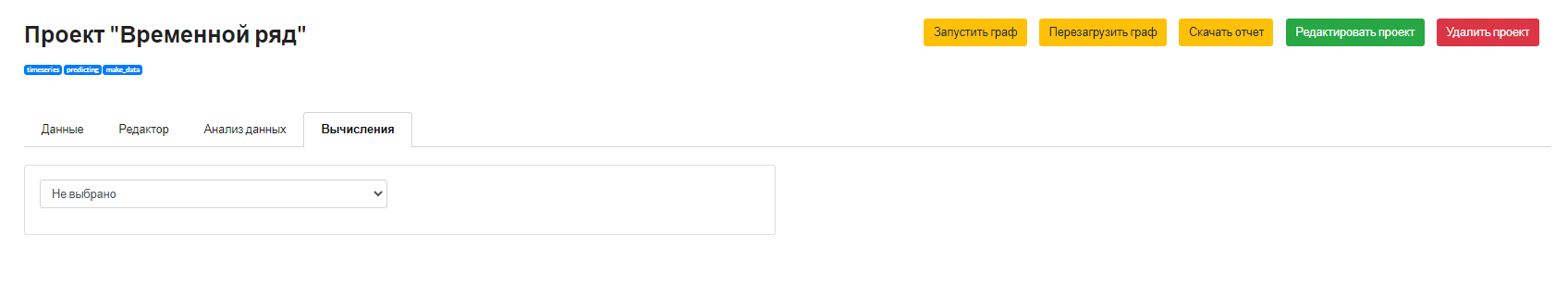


Рисунок 3.4.11 – Раздел «Вычисления» проекта анализа и прогнозирования временного ряда

* 1. Выбор узла с моделью, предсказывающей значения временного ряда на определенное пользователем количество дней вперед (рис. 3.4.12).

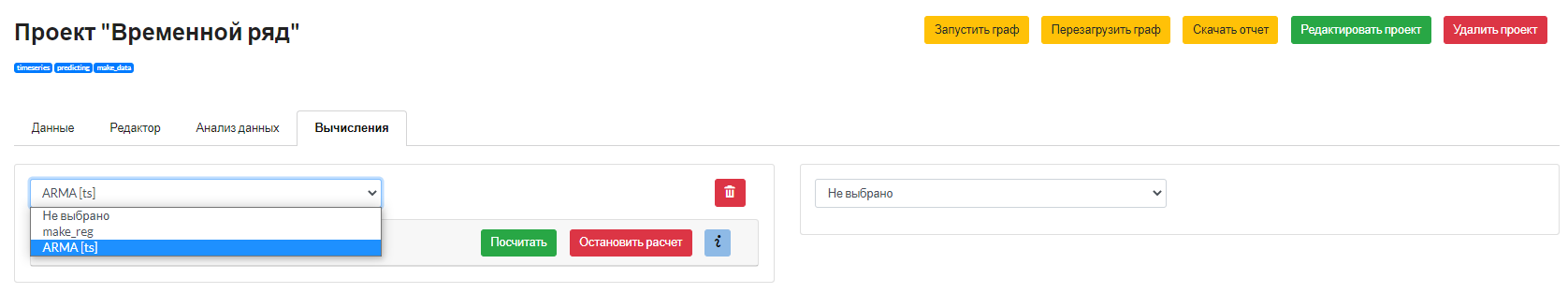


Рисунок 3.4.12 – Выбор узла с моделью, предсказывающей значения временного ряда на определенное пользователем количество дней вперед, в разделе «Вычисления»

* 1. Настройка модели, а именно установка параметров, по которым будет работать модель, предсказывающая значения временного ряда на определенное пользователем количество дней вперед, в меню «Расширенные настройки» (рис. 3.4.13).

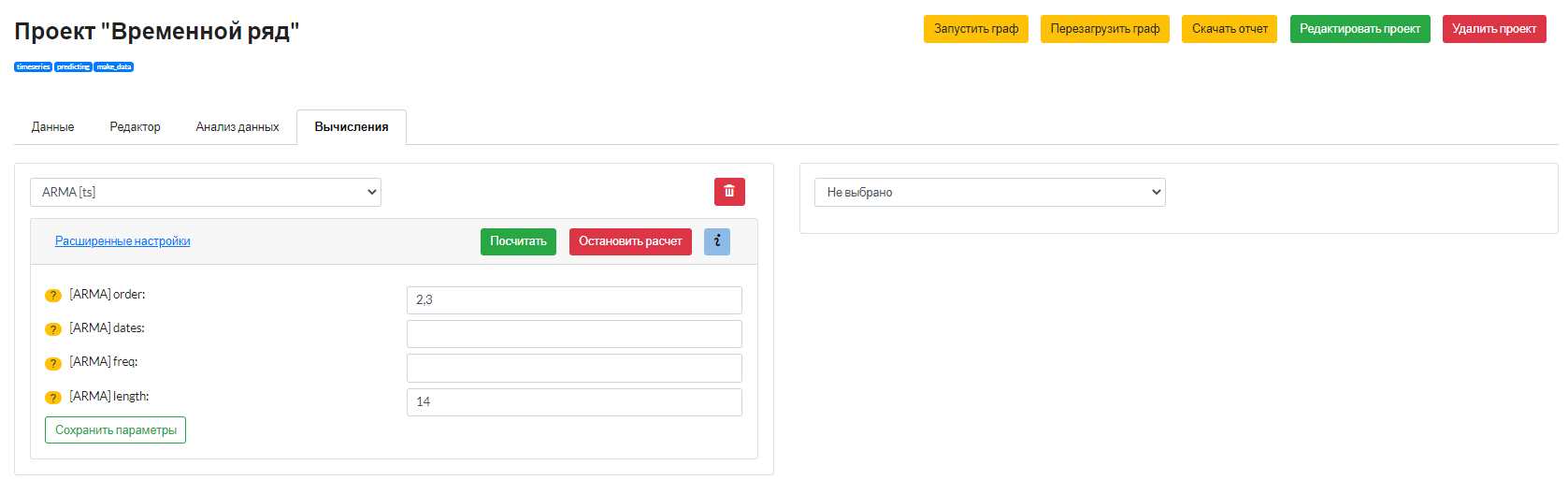


Рисунок 3.4.13 – Меню «Расширенные настройки» работать модели, предсказывающей значения временного ряда, в разделе «Вычисления»

* 1. Переход между вкладками с параметрами, полученными в результате работы модели, предсказывающей значения временного ряда на определенное пользователем количество дней вперед (рис. 3.1.14).

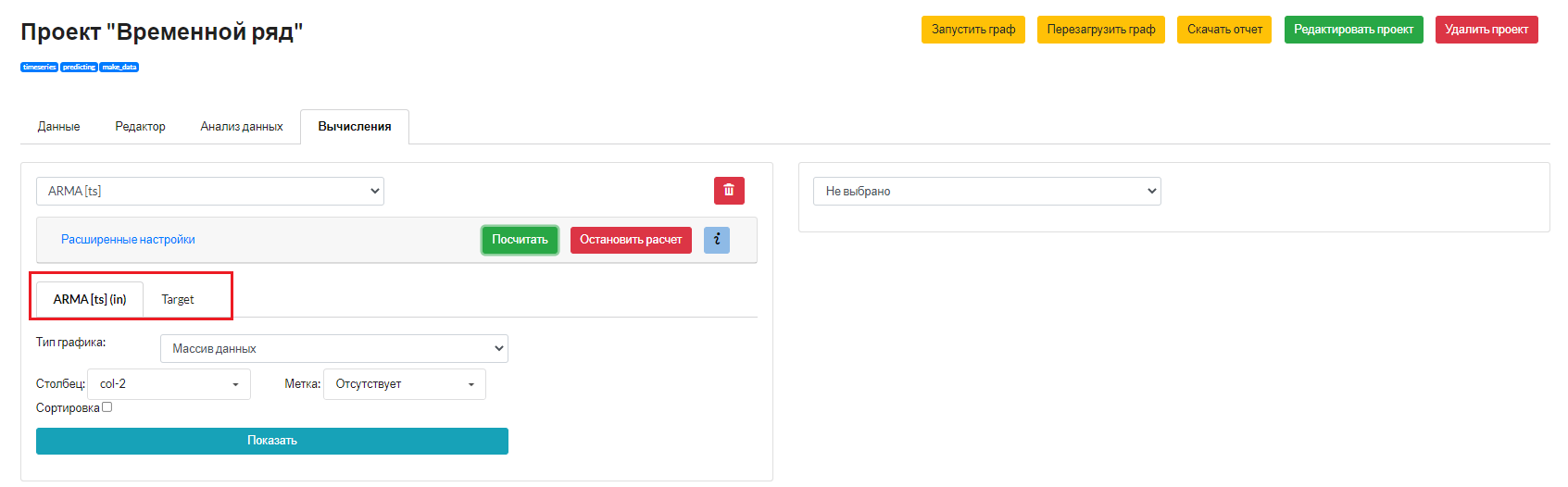


Рисунок 3.4.14 – Выбор вкладки с параметром, полученным в результате работы модели, предсказывающей значения временного ряда, в разделе «Вычисления»

* 1. Выбор типа визуализации (рис. 3.4.15), типа данных (3.4.16) и визуализация результатов работы модели (рис. 3.4.17 - 3.4.18).

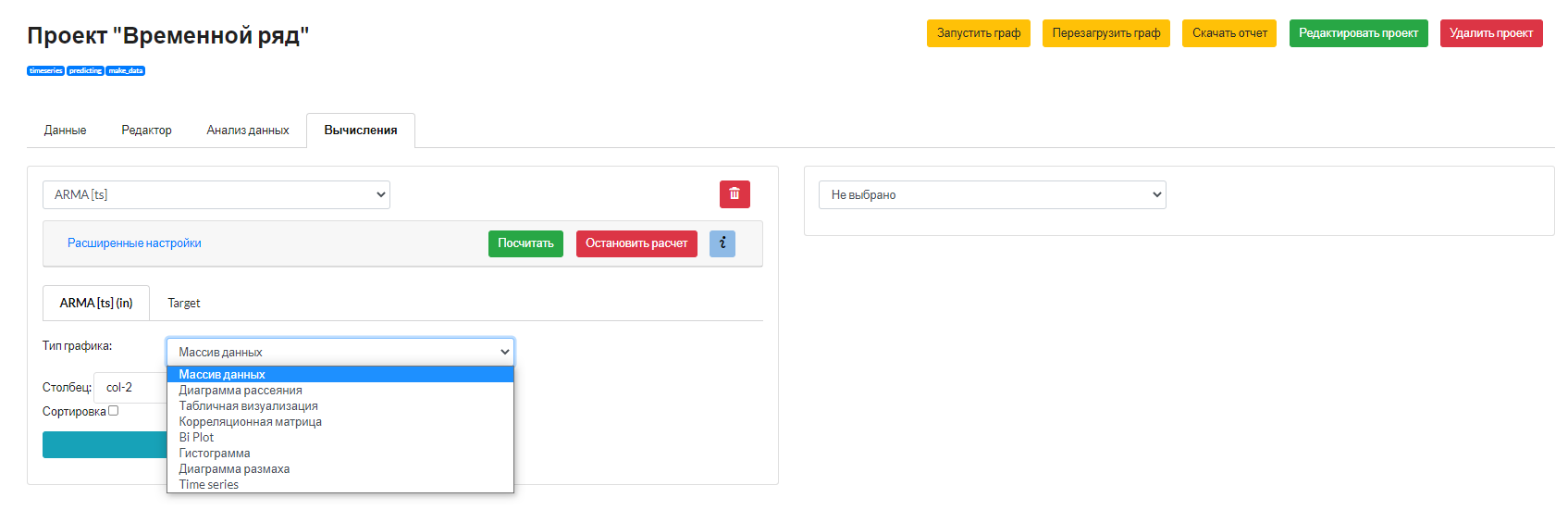


Рисунок 3.4.15 – Выбор типа визуализации результатов работы модели, предсказывающей значения временного ряда, в разделе «Вычисления»

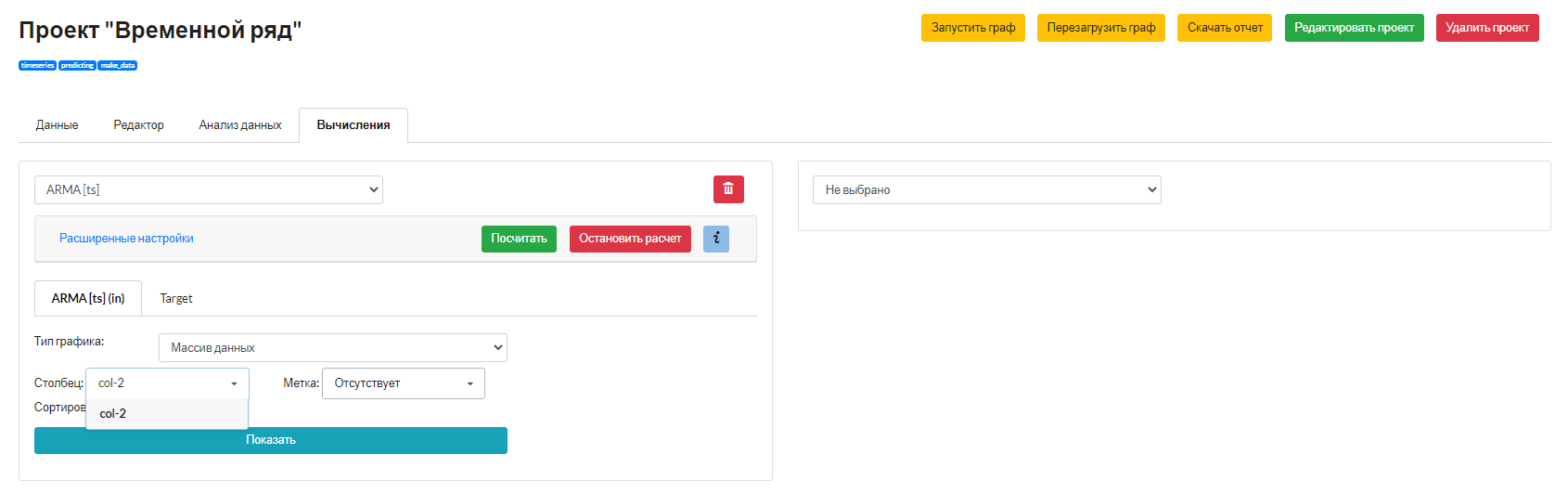


Рисунок 3.4.16 – Выбор типа данных результатов работы модели, предсказывающей значения временного ряда, в разделе «Вычисления»

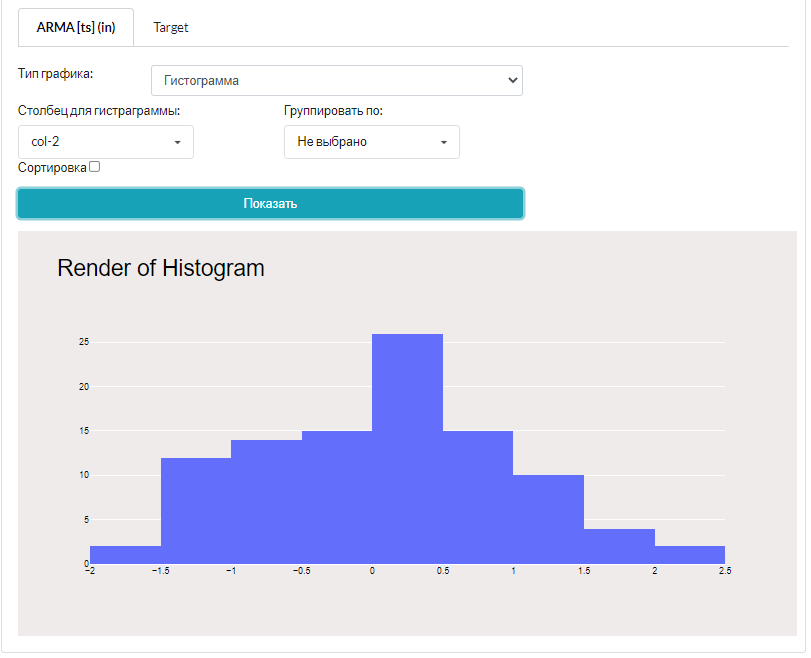


Рисунок 3.4.17 – Визуализация результатов работы модели, предсказывающей значения временного ряда, в разделе «Вычисления»

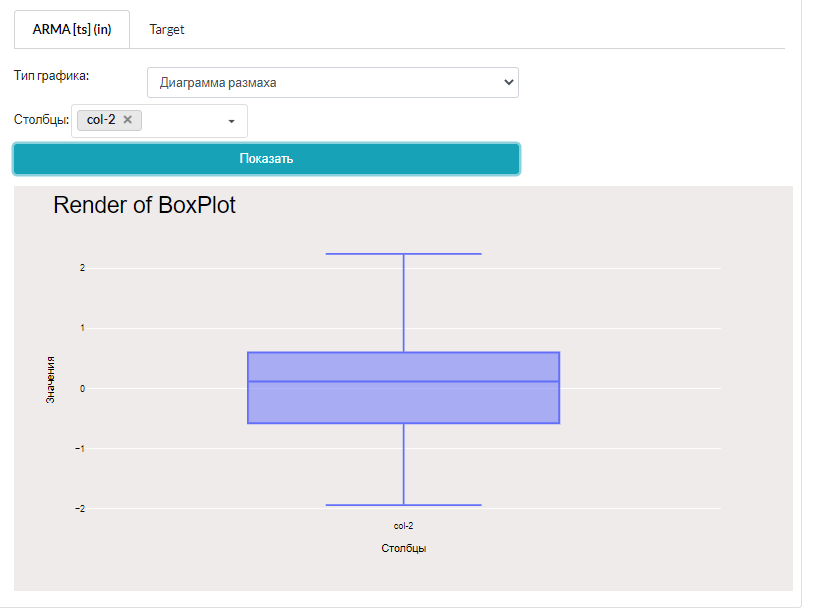


Рисунок 3.4.18 – Визуализация результатов работы модели, предсказывающей значения временного ряда, в разделе «Вычисления»

1. Сохранение результатов работы проекта:
   1. Загрузка результатов работы модели, содержащих предсказанные значения временного ряда на примере некоторого ряда данных, собранных в разные моменты времени, на ПК пользователя в разделе «Вычисления».
   2. Загрузка отчетов работы модели на ПК пользователя из:

* Раздела «Данные» (рис. 3.4.19)
* Раздела «Редактор»
* Раздела «Анализ данных»
* Раздела «Вычисления»

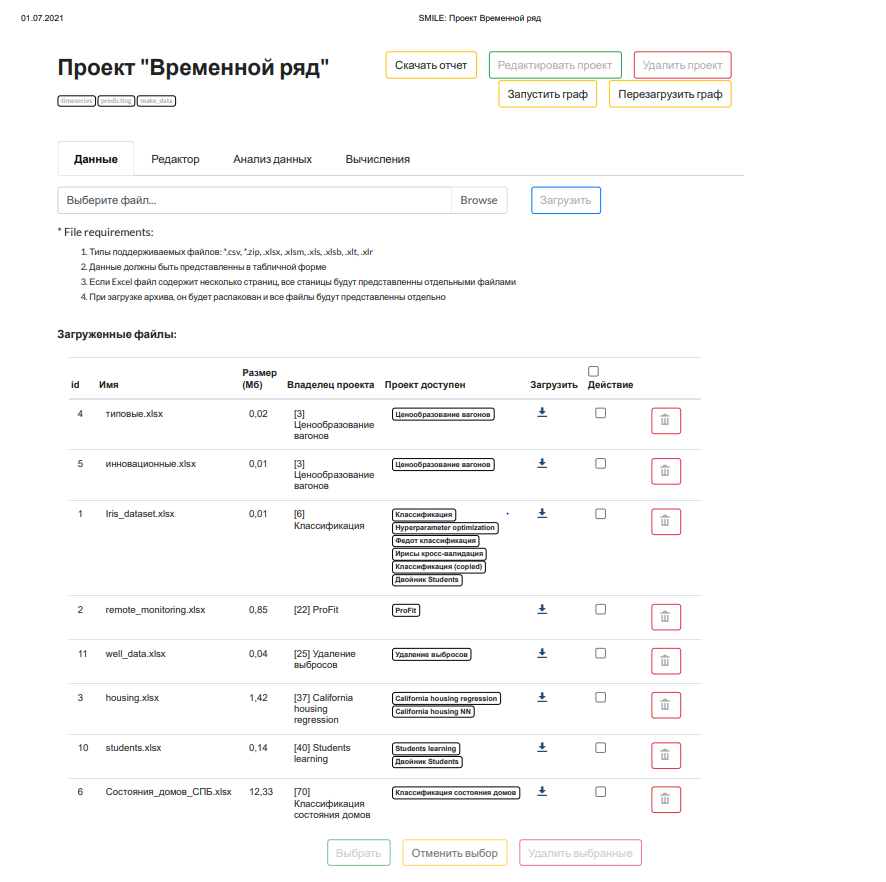


Рисунок 3.4.19 – Отчет раздела «Вычисления» проекта анализа и прогнощзирования временного ряда

**3.5. Оценка моделей (проект «Models evaluating»)**

Работа метода оценки моделей на примере оценки моделей классификации, кластеризации и линейной регрессии включает в себя:

1. Загрузку/выбор данных о решаемой задачи в разделе «Данные» (рис. 3.5.1).

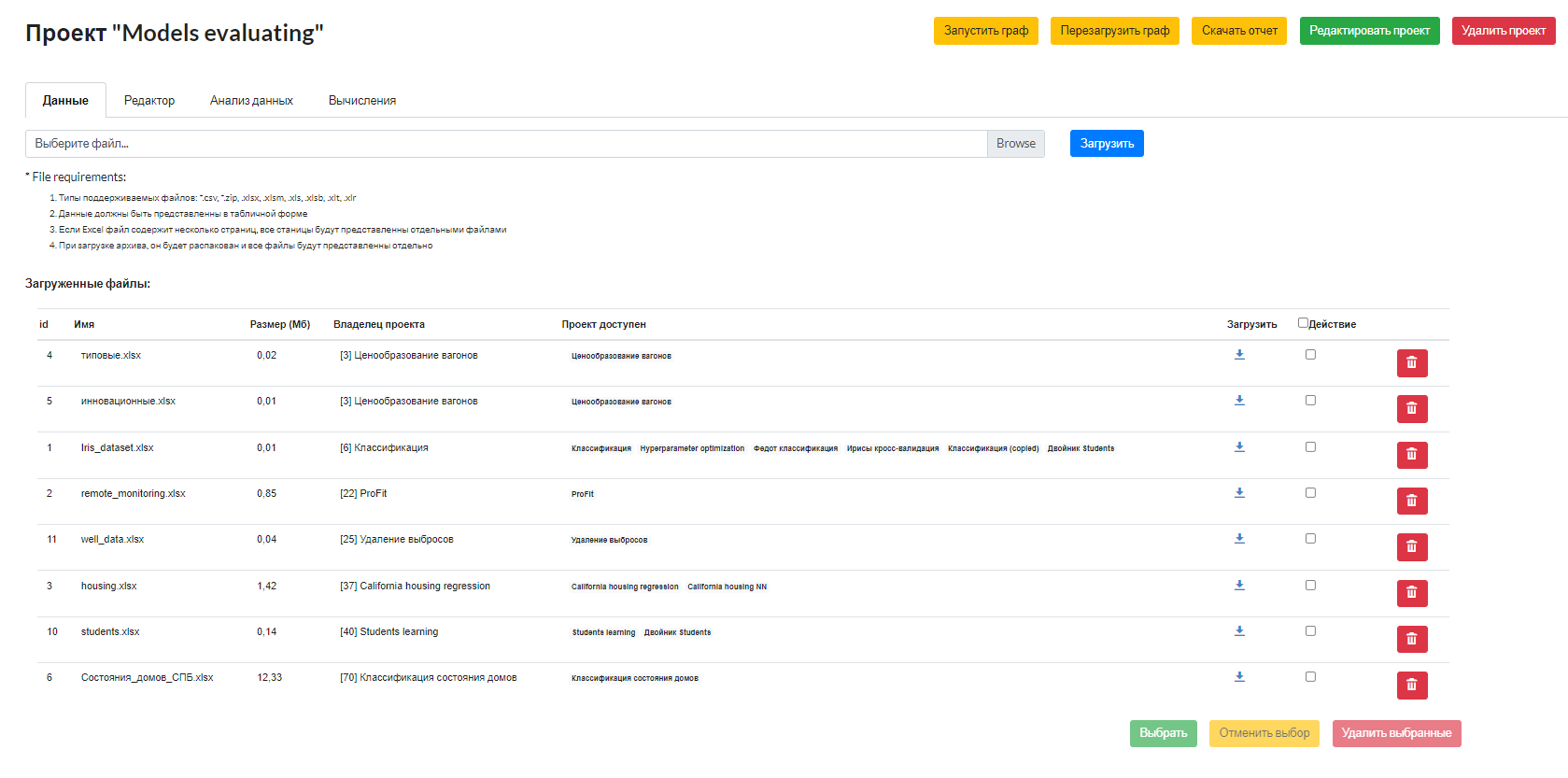


Рисунок 3.5.1 – Загрузка данных о ценах вагонов в разделе «Данные»

1. Работу с редактором графов, создаваемых для наглядного представления действий, осуществляемых с данными решаемой задачи, в разделе «Редактор» (рис. 3.5.2).

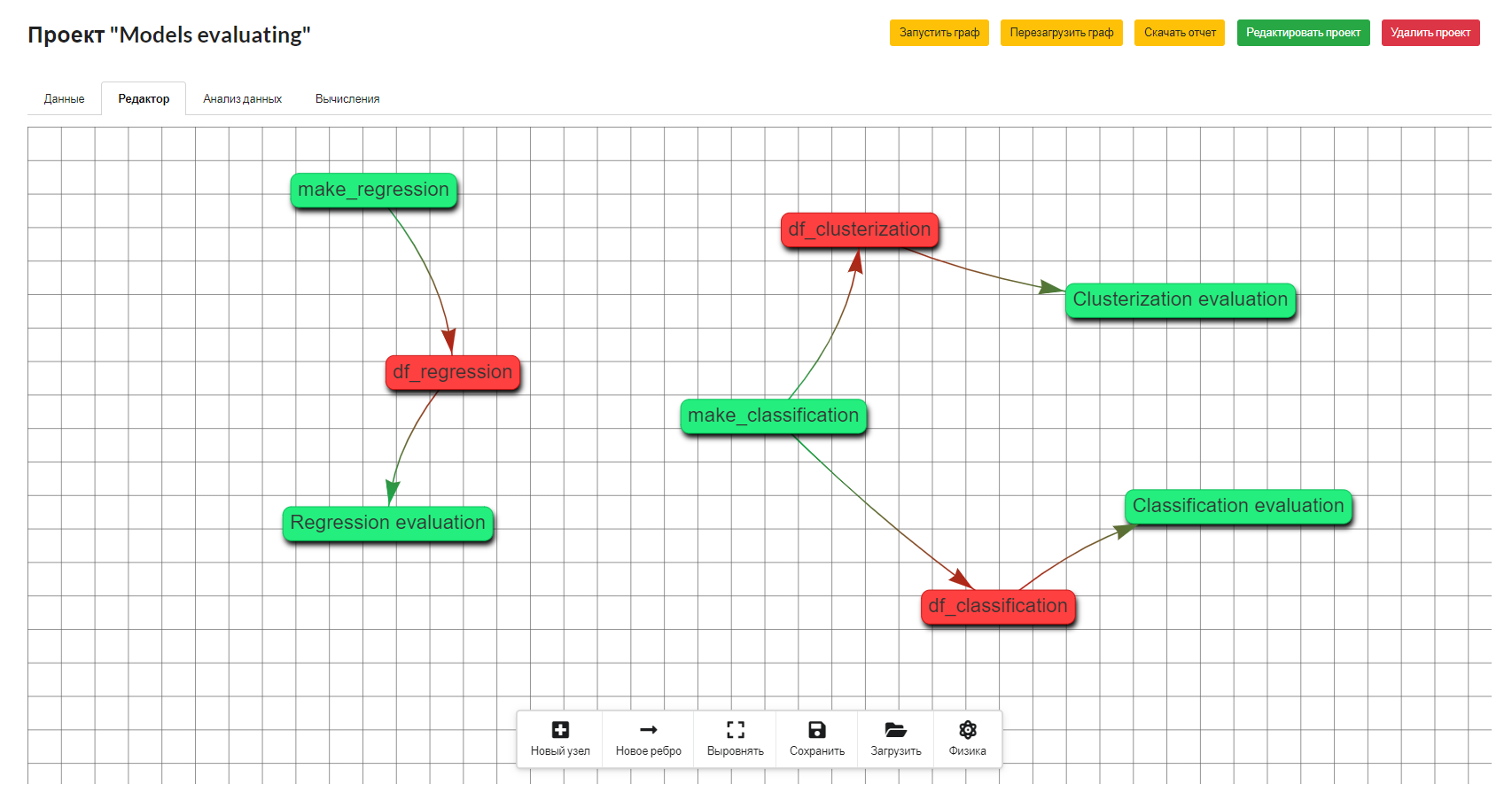


Рисунок 3.5.2 – Работа с редактором графа для данных решаемой задачи в разделе «Редактор»

* 1. Построение графа, а именно добавление узлов и ребер, которые используются для реализации цепи действий, осуществляемых с данными (рис. 3.5.3).

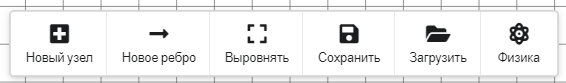


Рисунок 3.5.3 – Меню работы с графами в разделе «Редактор»

Цепь действий для задачи оценки моделей, используемых для решения задачи:

* Модель линейной регрессии, построенная на данных решаемой задачи (рис. 3.5.4)

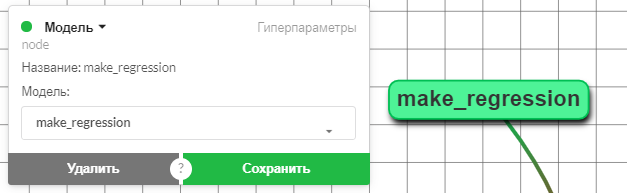


Рисунок 3.5.4 – Узел «Модель линейной регрессии» графа для данных решаемой задачи

* Массив с результатами работы модели линейной регрессии (рис. 3.5.5)

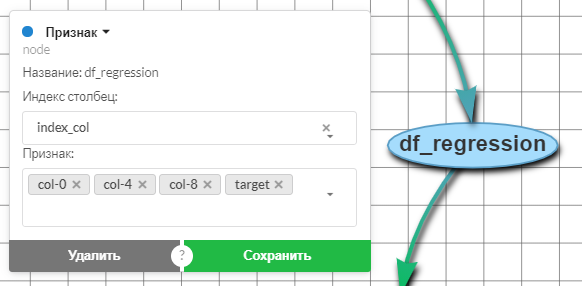


Рисунок 3.5.5 – Узел «Массив с результатами работы модели линейной регрессии» графа для данных решаемой задачи

* Оценка результатов работы модели линейной регрессии (рис. 3.5.6)

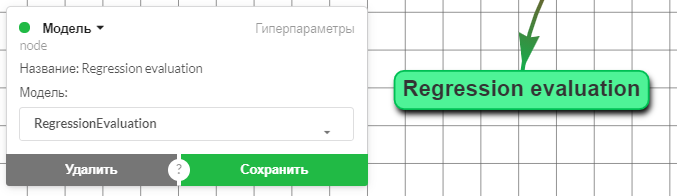


Рисунок 3.5.6 – Узел «Оценка результатов работы модели линейной регрессии» графа для данных решаемой задачи

* Модель классификации/кластеризации, построенная на данных решаемой задачи (рис. 3.5.7)

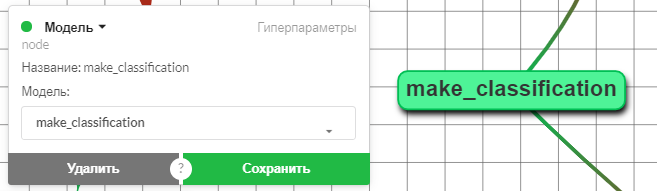


Рисунок 3.5.7 – Узел «Модель классификации/кластеризации» графа для данных решаемой задачи

* Массив с результатами работы модели кластеризации (рис. 3.5.8)

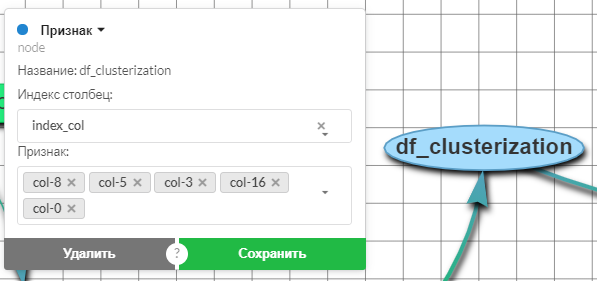


Рисунок 3.5.8 – Узел «Массив с результатами работы модели кластеризации» графа для данных решаемой задачи

* Оценка результатов работы модели кластеризации (рис. 3.5.9)

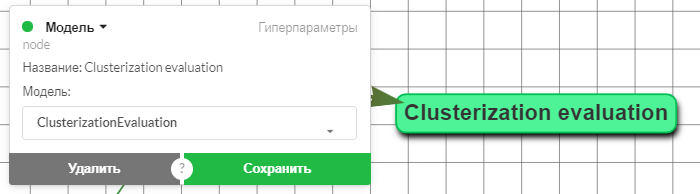


Рисунок 3.5.9 – Узел «Оценка результатов работы модели кластеризации» графа для данных решаемой задачи

* Массив с результатами работы модели классификации (рис. 3.5.10)



Рисунок 3.5.10 – Узел «Массив с результатами работы модели классификации» графа для данных решаемой задачи

* Оценка результатов работы модели классификации (рис. 3.5.11)

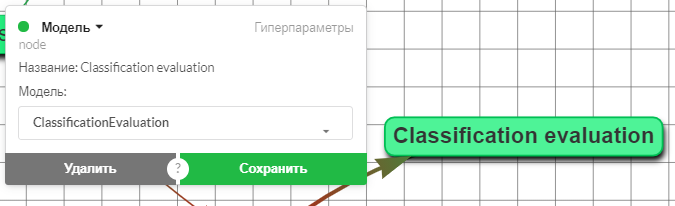


Рисунок 3.5.11 – Узел «Оценка результатов работы модели классификации» графа для данных решаемой задачи

* 1. Выбор свойств узлов/ребер (для ошибок валидации – красные узла, для признаков - синие узлы, а для моделей - зеленые узлы). Признаками (ошибками валидации до запуска модели) в примере являются «Массив с результатами работы модели линейной регрессии», «Массив с результатами работы модели кластеризации» и «Массив с результатами работы модели классификации», а моделями – «Модель линейной регрессии, построенная на данных решаемой задачи», «Модель классификации/кластеризации, построенная на данных решаемой задачи», «Оценка результатов работы модели линейной регрессии», «Оценка результатов работы модели кластеризации» и «Оценка результатов работы модели классификации» (рис. 3.5.12).

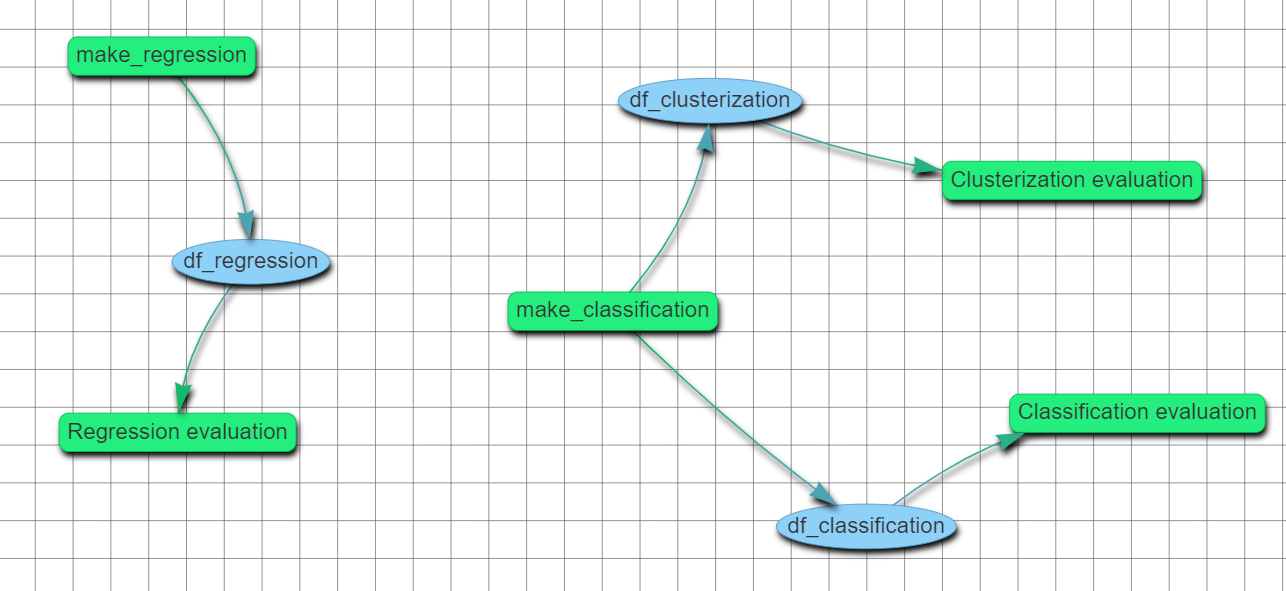


Рисунок 3.5.12 – Граф для данных по вагонам в разделе «Редактор»

1. Процесс просмотра входных данных проекта оценки моделей, в разделе «Анализ данных» (рис. 3.5.13-3.5.14).

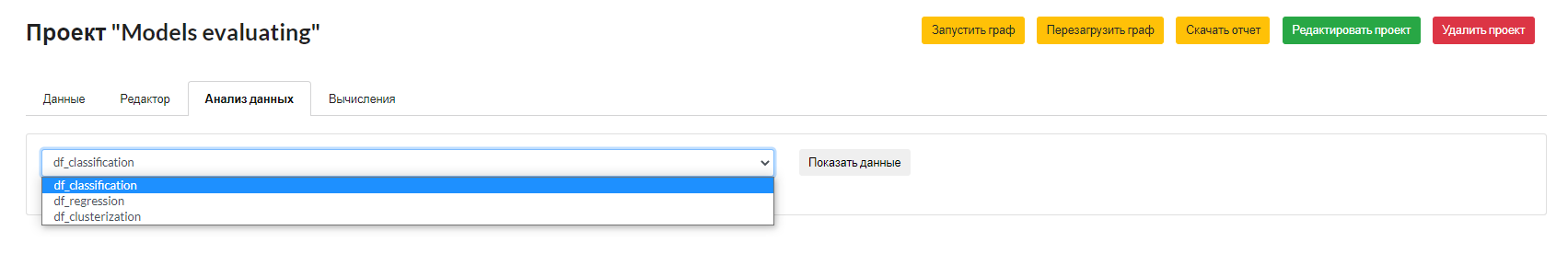


Рисунок 3.5.13 – Раздел «Анализ данных» проекта оценки моделей

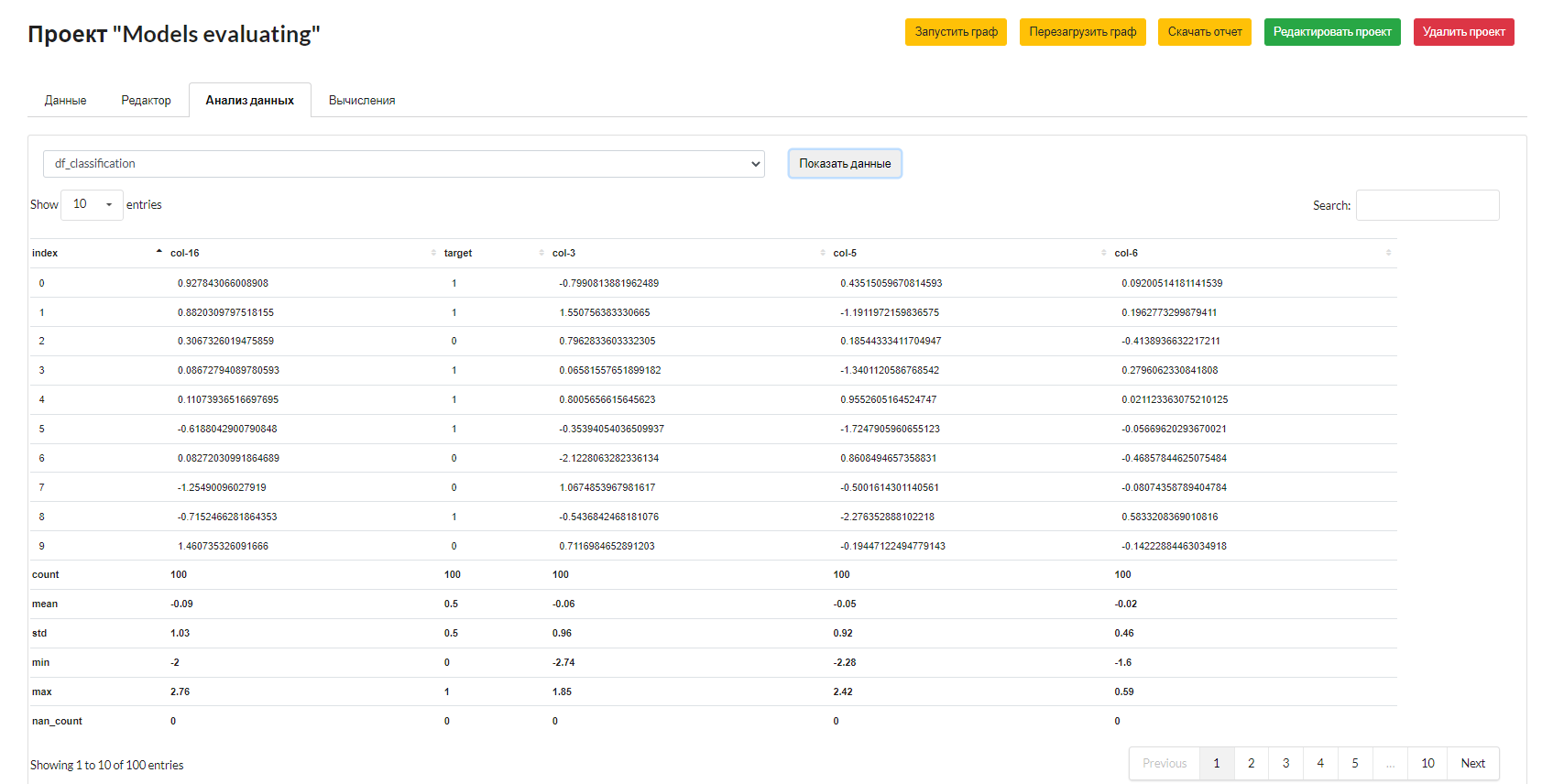


Рисунок 3.5.14 – Просмотр входных данных проекта оценки моделей

1. Процесс вычисления, осуществляемого моделями линейной регрессии, кластеризации и классификации, а также оценка их результативности по отношению к решаемой задаче, в разделе «Вычисления» (рис. 3.5.15).

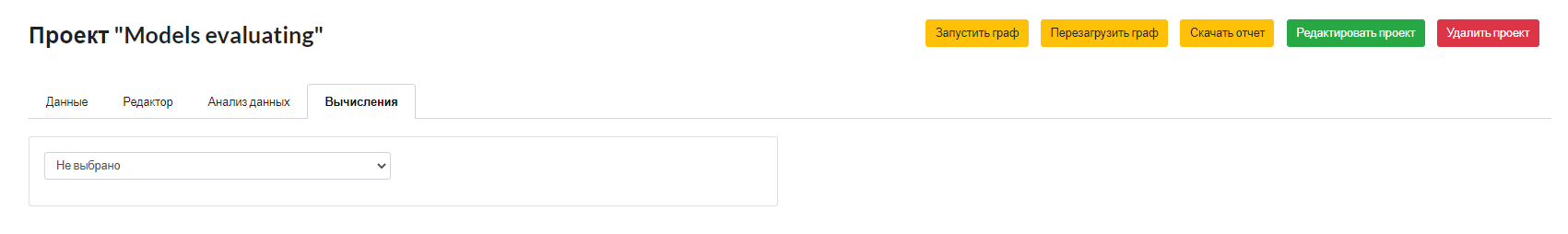


Рисунок 3.5.15 – Раздел «Вычисления» проекта оценки моделей

* 1. Выбор узла с моделью оценки результатов работы кластеризации (рис. 3.5.16).

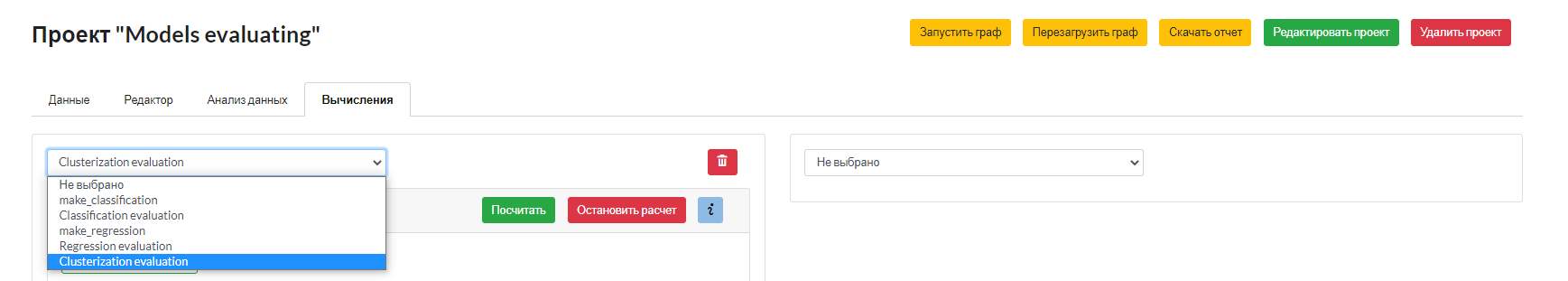


Рисунок 3.5.16 – Выбор узла с моделью оценки результатов работы кластеризации в разделе «Вычисления»

* 1. Настройка модели, а именно установка параметров, по которым будет выполнена классификация/кластеризация, в меню «Расширенные настройки» (рис. 3.5.17).

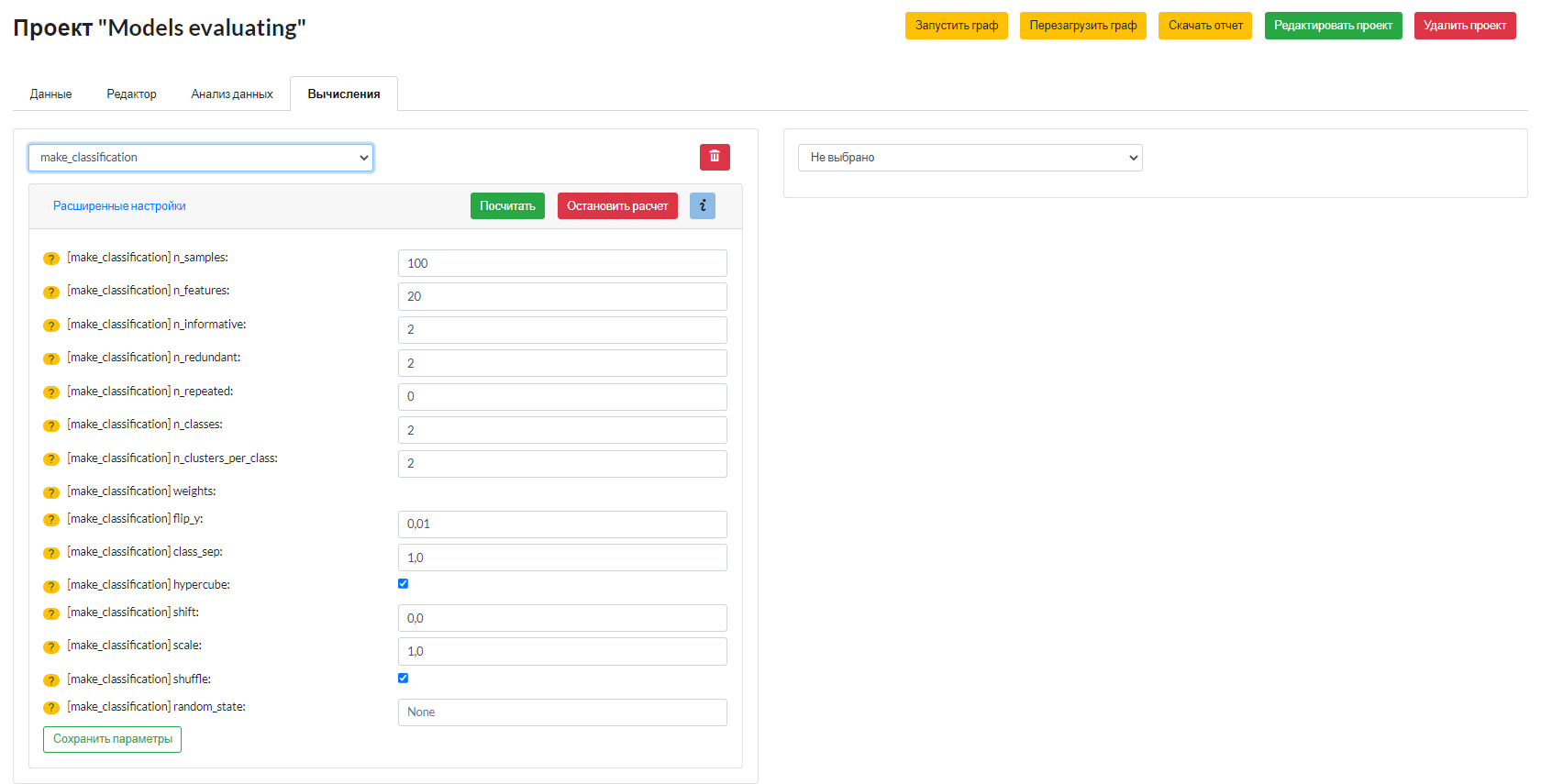


Рисунок 3.5.17 – Меню «Расширенные настройки» модели классификации/кластеризации для решаемой задачи в разделе «Вычисления»

* 1. Переход между вкладками с параметрами, полученными в результате работы модели оценки результатов работы кластеризации решаемой задачи (рис. 3.1.18).

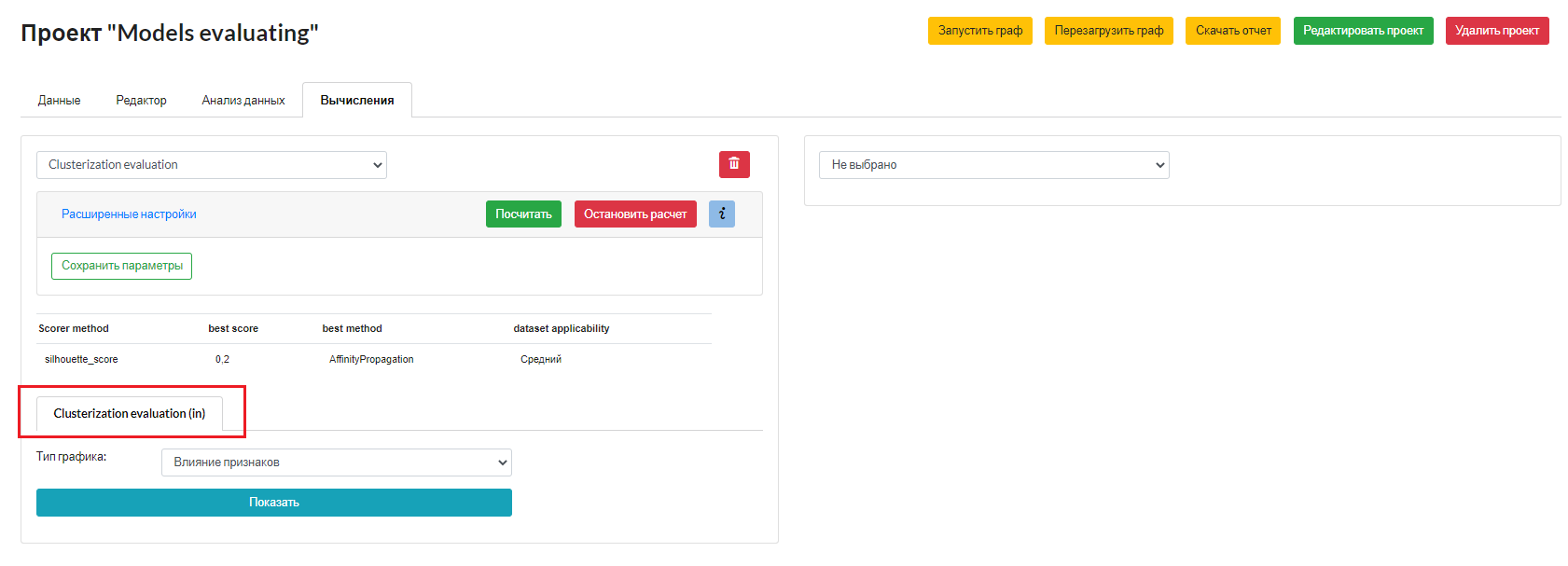


Рисунок 3.5.18 – Выбор вкладки с параметром, полученным в результате работы оценки результатов работы кластеризации решаемой задачи в разделе «Вычисления»

* 1. Выбор типа визуализации (рис. 3.5.19), типа данных (3.5.20) и визуализация результатов работы моделей (рис. 3.5.21).

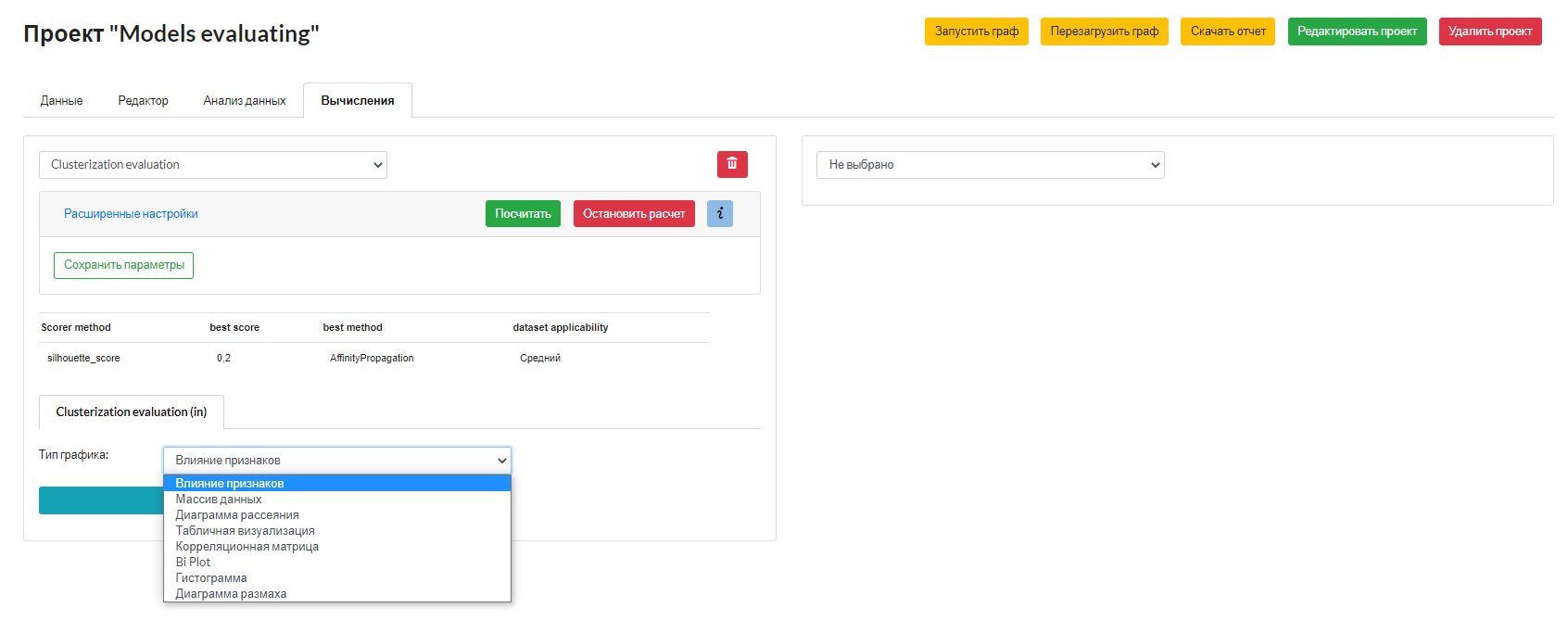


Рисунок 3.5.19 – Выбор типа визуализации результатов работы модели оценки результатов работы кластеризации решаемой задачи в разделе «Вычисления»

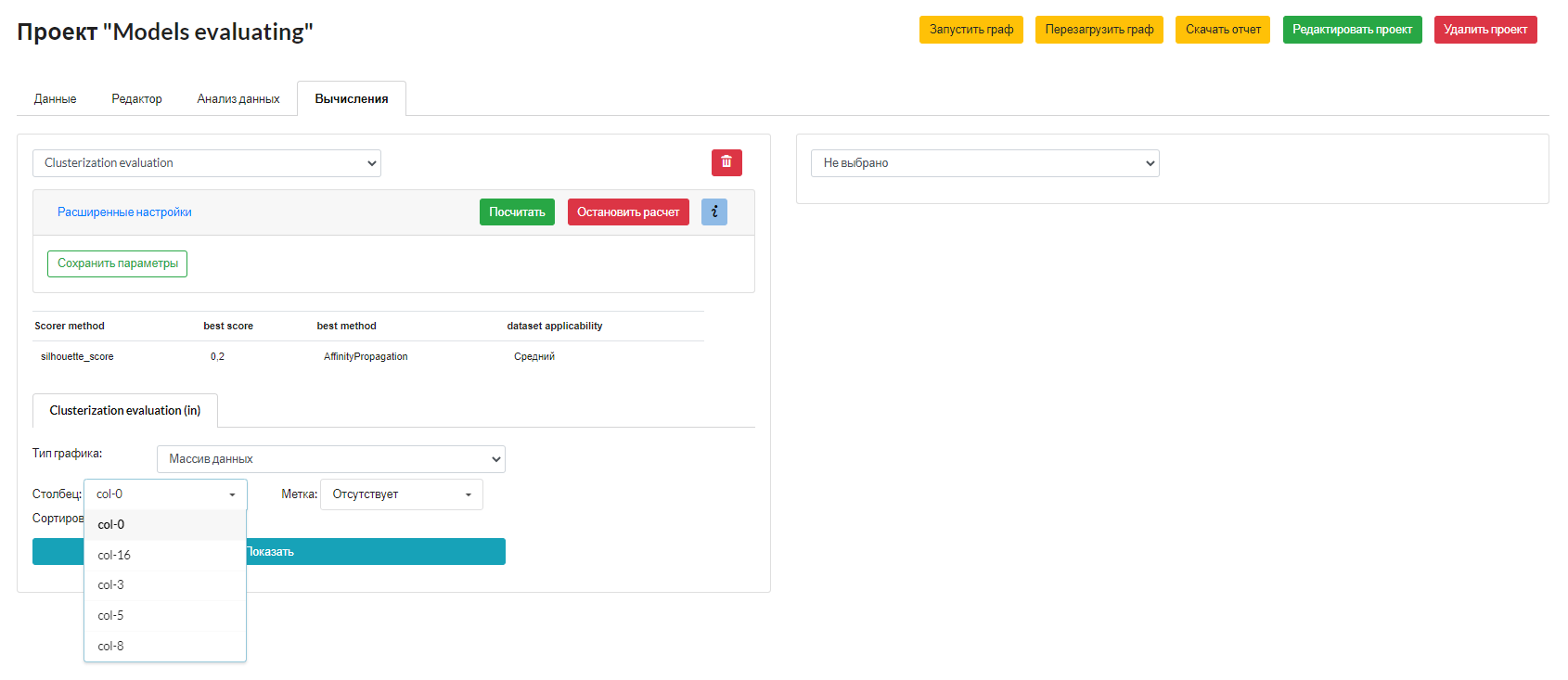


Рисунок 3.5.20 – Выбор типа данных результатов работы модели оценки результатов работы кластеризации решаемой задачи в разделе «Вычисления»

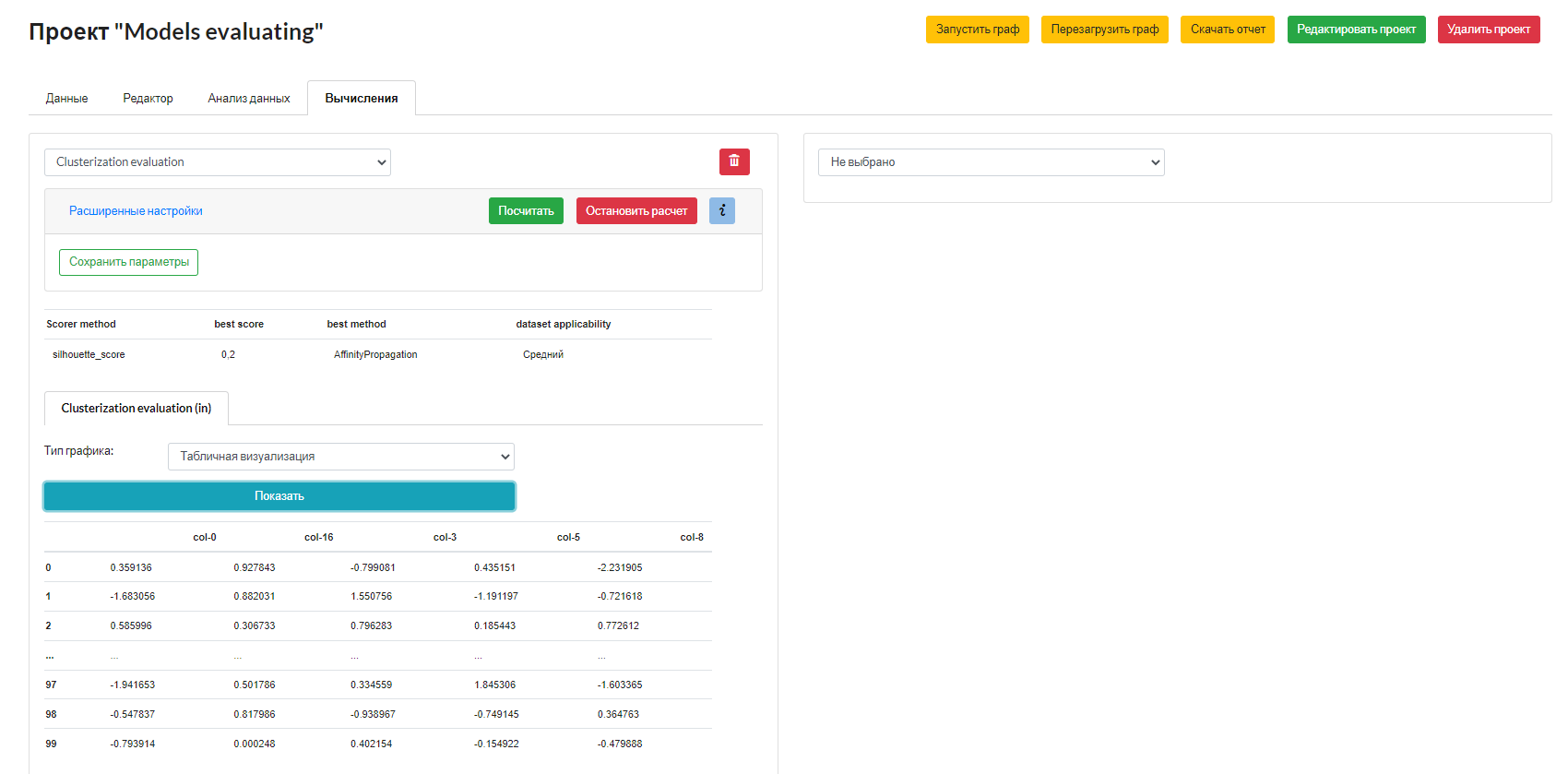


Рисунок 3.5.21 – Визуализация результатов работы модели оценки результатов работы кластеризации в разделе «Вычисления»

1. Сохранение результатов работы проекта:
   1. Загрузка результатов работы модели, содержащих данные выполнения моделей кластеризации или классификации/кластеризации, на ПК пользователя в разделе «Вычисления» (рис. 3.5.22).

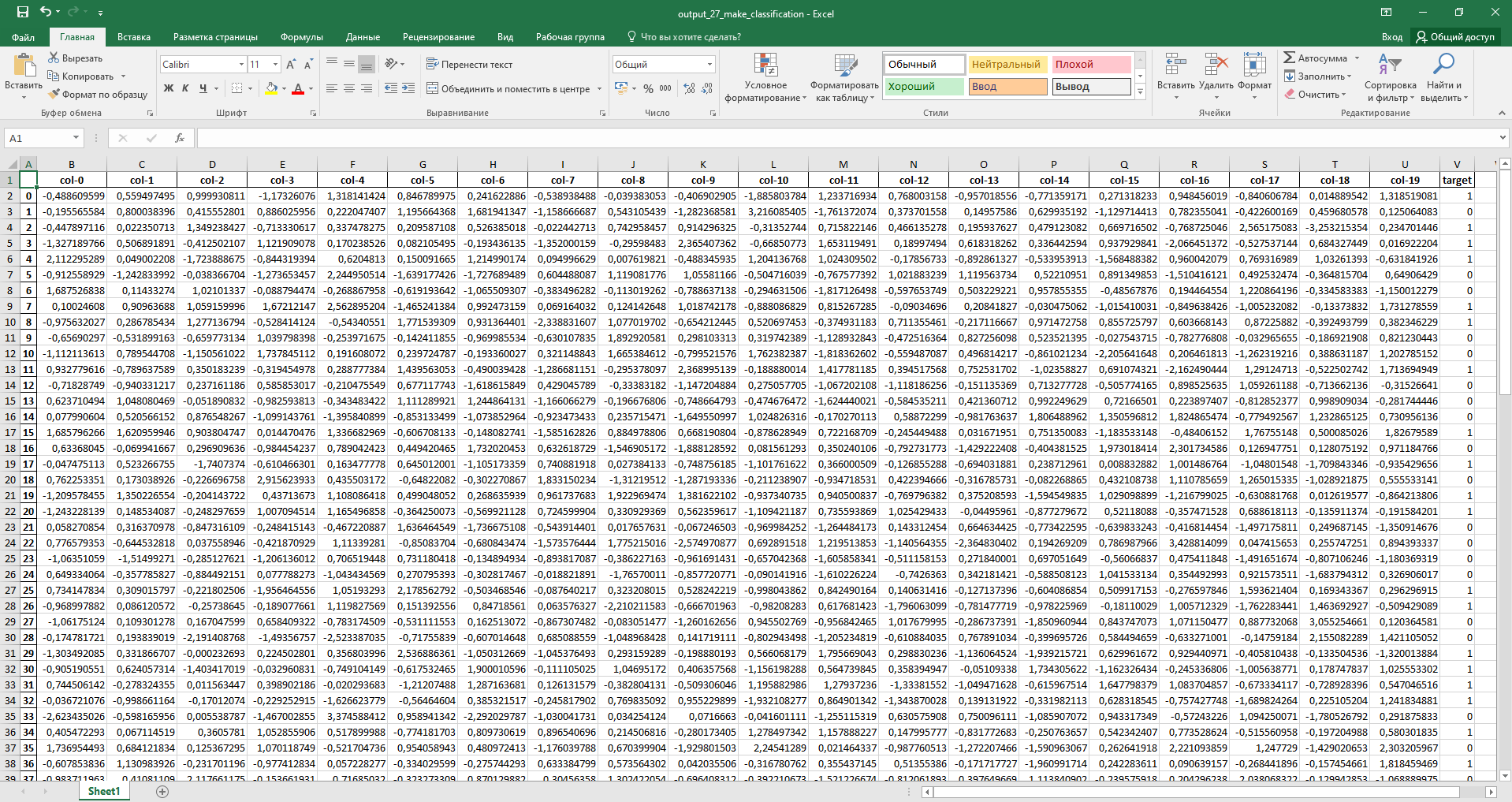


Рисунок 3.5.22 – Файл с результатами работы модели классификации/кластеризации в разделе «Вычисления» на ПК пользователя

* 1. Загрузка отчетов работы модели на ПК пользователя из:
* Раздела «Данные»
* Раздела «Редактор»
* Раздела «Анализ данных»
* Раздела «Вычисления» (рис. 3.5.23)

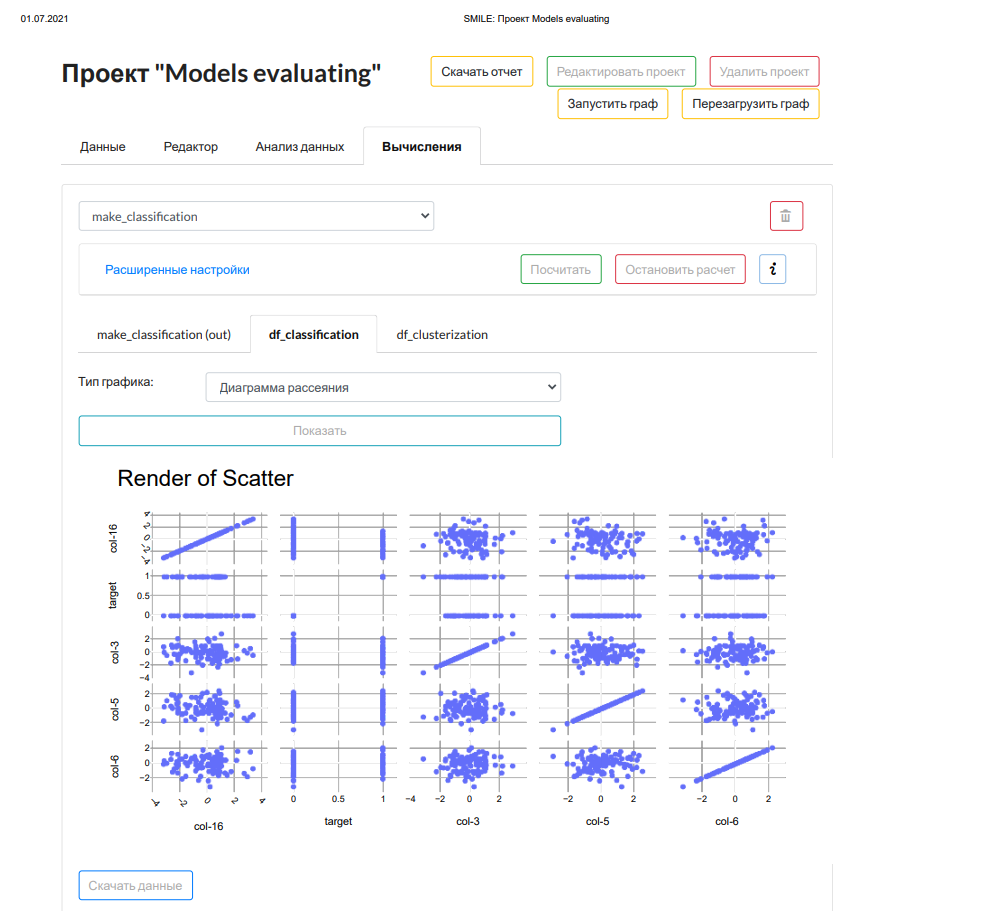


Рисунок 3.5.23 – Отчет раздела «Вычисления» проекта оценки моделей

1. **Входные и выходные данные**
   1. **Сведения о входных данных**
      1. **Данные, загружаемые в проекты**

Платформа работает:

* На основе данных, вводимых пользователем в интерфейсе программы в разделе «Вычисления» проектов (ввод параметров модели в меню «Расширенные настройки» раздела «Вычисления» редакторов проектов);
* На основе файлов, загружаемых через интерфейс платформы (в разделе «Данные» редактора проектов) (см. рис. 4.1.1.1):

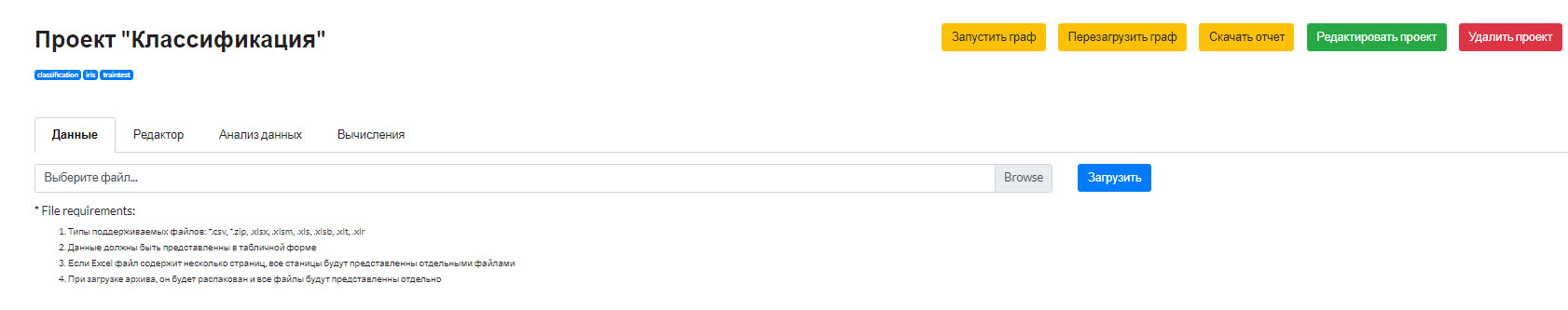


Рисунок 4.1.1.1 – Интерфейс загрузки входных файлов

При этом требования, предъявляемые к загружаемым файлам с данными прописаны под полем с наименованием загружаемого файла (рис. 4.1.1.2).

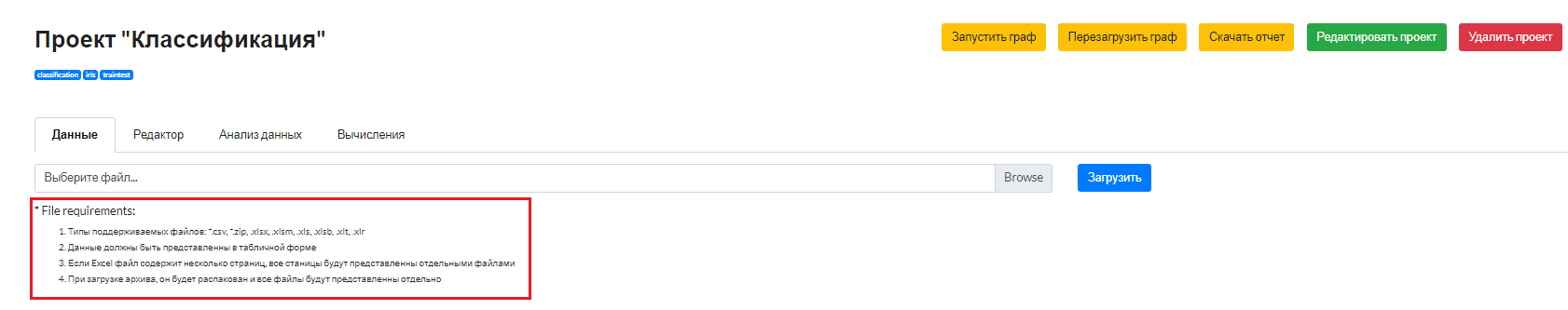


Рисунок 4.1.1.2 – Требования к загружаемому файлу

Данные в файлах должны быть представлены в виде таблицы, заголовок которой содержит наименования атрибутов, а ниже в колонках расположены значения каждого атрибута (рис 4.1.1.3).

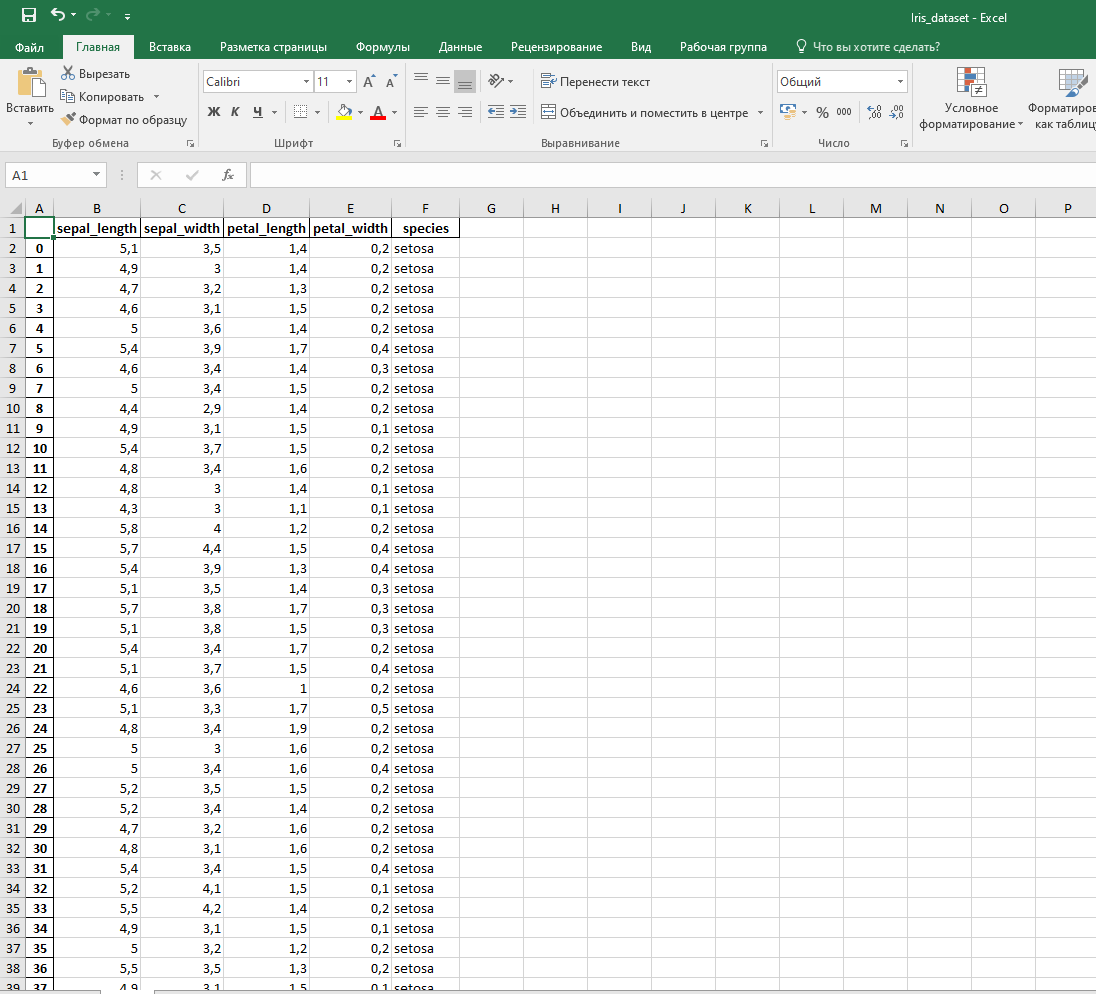


Рисунок 4.1.1.3 – Загружаемый в проект файл «Iris\_dataset.xlsx» на примере данных проекта «Классификация»

При этом должны быть соблюдены следующие требования к загружаемым файлам:

1. Типы загружаемых файлов могут быть только : csv, zip, xlsx, .xlsm, .xls, .xlsb, .xlt и.xlr. При этом, если файл Microsoft Excel содержит несколько страниц, то каждая страница будет загружена в виде отдельного файла. Также при загрузке архива, все файлы архива будут загружены по отдельности.
2. Значения колонки, содержащей дату, должны быть записаны в следующем формате: YYYY-mm.
3. Ячейки не должны быть пустыми, за исключением ячеек колонки предсказываемой величины.
   * 1. **Создание пользовательских моделей**

Для создания собственной модели, работающей в программе, существуют справочные материалы, а именно пример модели и требования к создаваемой модели. Для загрузки примера модели необходимо нажать на «Скачать пример», а для загрузки требований к модели – на «Скачать требования» (рис. 4.1.2.1).

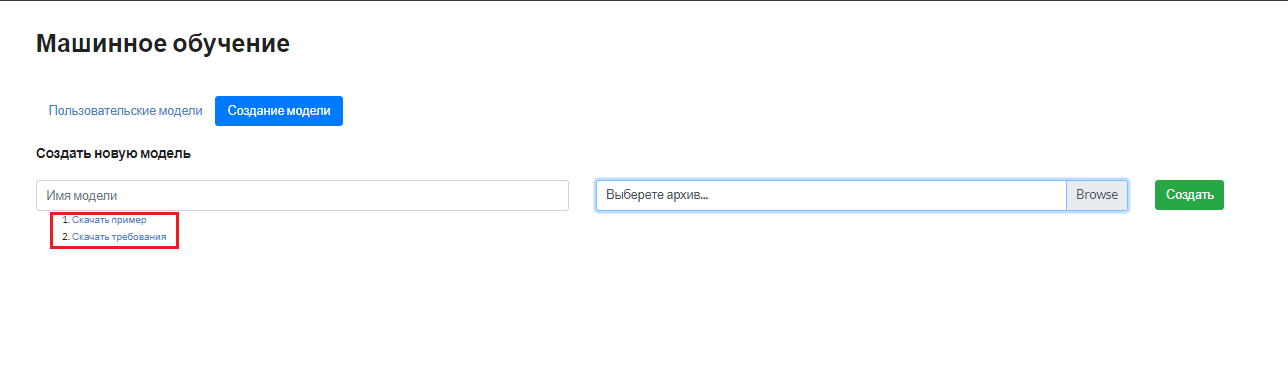


Рисунок 4.1.2.1 – Загрузка примера модели и требований к создаваемой модели в подразделе «Пользовательские модели» редактора свойств проектов и графов проектов типа «Балансовые модели»

Пример модели представляет собой сжатую ZIP-папку с наименованием «example.zip», а требования – текстовый документ с наименованием «Требования для Машинное обучение.txt» (рис. 4.1.2.2). Новая модель, а именно её характеристики и алгоритм работы, прописываются с использованием языка программирования Python. В листинге 4.1.2.1 представлен пример новой модели, скачанный из подраздела «Создание модели» и расположенный в ZIP-папку с наименованием «example.zip», в файле «\_abstract\_module.py».

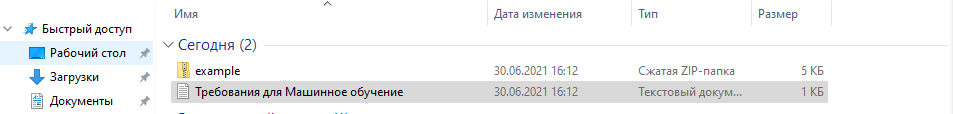


Рисунок 4.1.2.2 – Справочные материалы для моделей проектов в папке загрузок ПК пользователя

Листинг 4.1.2.1

|  |
| --- |
| import ast  import re  from abc import ABC, abstractmethod  import numpy as np  from django.template import Template, Context  from numpydoc.docscrape import NumpyDocString  from data.modules.machine\_learning.plot\_methods.dataframe import dataframe  from data.modules.machine\_learning.plot\_methods.histogram import histogram  from data.modules.machine\_learning.plot\_methods.bi\_plot import bi\_plot  from data.modules.machine\_learning.plot\_methods.correlation\_matrix import correlation\_matrix  from data.modules.machine\_learning.plot\_methods.table\_visualiser import table\_visualiser  from data.modules.machine\_learning.plot\_methods.scatter import scatter  from data.modules.machine\_learning.plot\_methods.box\_plot import box\_plot    def prepare\_model\_arguments(arguments):  parameters = {}  for k, v in arguments.items():  if isinstance(v, str) and not v:  continue  if isinstance(v, str) and ';' in v:  if v.count(';') == 2:  min\_val, max\_val, step = v.split(';')  else:  min\_val, max\_val = v.split(';')  step = '1'  v = np.arange(ast.literal\_eval(min\_val), ast.literal\_eval(max\_val) + 1, ast.literal\_eval(step)).tolist()  elif isinstance(v, str) and v and v[0] == '(' and v[-1] == ')': # it is tuple  v = tuple(map(ast.literal\_eval, filter(lambda v: v, v[1:-1].split(','))))  elif isinstance(v, str) and v.replace('.', '').isnumeric():  v = float(v) if '.' in v else int(v)  parameters[k] = None if isinstance(v, str) and v.lower() in {'none', 'nan'} else v  return parameters  def process\_arguments(f):  def func(self, \*args, \*\*kwargs):  if len(args) == 0 and 'properties' not in kwargs:  return f(self, \*args, \*\*kwargs)  parameters\_input = kwargs['properties'] if 'properties' in kwargs else args[2]  parameters = prepare\_model\_arguments(parameters\_input)  if 'properties' in kwargs:  kwargs['properties'] = parameters  return f(self, \*args, \*\*kwargs)  return f(self, \*args[:2], parameters, \*\*kwargs)  return func  def get\_params(name, params):  res = {}  for k, v in params.items():  k\_splitted = k.split('.')  if name == k\_splitted[0] and k\_splitted[1] not in res:  if len(k\_splitted) == 3 and k\_splitted[2] in {'from', 'to'}:  param\_key = '{}.{}'.format(k\_splitted[0], k\_splitted[1])  val = [params['{}.from'.format(param\_key)], params['{}.to'.format(param\_key)]]  if val and val[0] and val[1]:  res[k\_splitted[1]] = val  elif isinstance(v, (bool, int, float)) or v:  res[k\_splitted[1]] = v  return res  class AbstractModule(ABC):  def \_\_init\_\_(self, instance, plots=None):  """  :param instance: Base instance method, it this class can be applied for several methods  :param plots:  """  self.instance = instance  self.params\_exceptions = {'verbose', 'n\_jobs', 'X', 'y'}  self.\_observers = []  self.separate\_thread = True  self.to\_plot = {'in': True, 'out': True}  self.plots\_model = {}  if plots is None:  plots = []  for visualizer in [dataframe, scatter, table\_visualiser, correlation\_matrix, bi\_plot, histogram, box\_plot]:  if visualizer not in plots:  plots.append(visualizer)  self.plots = dict(map(lambda v: (re.sub(r'([A-Z])', ' \\1', v.\_\_class\_\_.\_\_name\_\_).lstrip(), v), plots))  def initialize(self, node):  pass  def add\_observer(self, observer):  self.\_observers.append(observer)  def notify(self, \*\*kwargs):  for observer in self.\_observers:  observer.inside\_update(\*\*kwargs)  def get\_docs(self, docs=None):  """  [Optional if there is the instance]  Returns the documentation of instance method for creating of arguments for this module  :return:  """  if docs is None and self.instance is not None:  docs = self.instance.\_\_doc\_\_  if isinstance(docs, str) and docs.split('\n')[0]:  docs = '\n ' + docs  return docs  def get\_params(self, properties, only\_read=False, extra\_parameters: list = None) -> list:  """  Получить из документации кокретного метода/модели список параметров и информацию о них  :param properties: Сохраненные в бд свойства  :param only\_read: Если треубется запретить изменения возвращаемым свойствам (например, если зашел  пользователь в проект, у которого нет прав на редактирование)  :param extra\_parameters: Кастомизированные параметры выбранной модели (исключают параметры из документации  :return: Список объектов с параметрами  """  if self.instance is None:  return []  name, doc = self.get\_name(), self.get\_docs()  params\_exceptions: set = self.params\_exceptions  if extra\_parameters is not None:  params\_exceptions = params\_exceptions | set(map(lambda v: v['name'], extra\_parameters))  if doc is None:  return []  from data.modules.machine\_learning.properties.models\_title import get\_model\_parameters  parameters = get\_model\_parameters(name, NumpyDocString(doc)['Parameters'], properties, only\_read)  return list(filter(lambda v: v['name'] not in params\_exceptions, parameters))  def get\_extra\_params(self, extra\_params=None, \*\*kwargs) -> list:  """  Вернуть дополнительные параметры для модели  [Optional] Extension of model parameters  :param extra\_params: list(dict(model: str, name: str, default: str, description: str, types: dict))  :param kwargs: dict(node: Node, default\_values: dict)  :return:  """  only\_read = kwargs.get('only\_read', False)  if extra\_params is None:  extra\_params = []  if 'prop\_values' in kwargs:  prop\_values = get\_params(self.get\_name(), kwargs['prop\_values']) # Получить свойства для модуля  for param in extra\_params:  param['value'] = prop\_values.get(param['name'], param['default'])  param['description'] = f'{param.get("description")}, default\_value = {param["default"]}'  if only\_read:  param['disabled'] = 'disabled'  return extra\_params  def get\_extra\_features(self, \*\*kwargs):  """  Получить дополнительные признаки для детей  [Optional] The values which the model to translate to out nodes  :param kwargs: columns=features, node=self, parents=parents, properties=properties  :return: dict(id\_of\_feature: dict(group=self.get\_name(), content=name\_if\_feature))  """  return {}  def collect\_parent\_features(self, node):  features = []  for edge in node.get\_edges('input'):  parent = node.graph.nodes\_map[edge['from']]  parent\_features: dict = parent.get\_feature\_options\_filtered(edge=edge)  features.extend(list(map(lambda v: v['content'], parent\_features.values())))  return features  def form\_features\_by\_list(self, node\_id: str, features: list, group\_name='') -> dict:  """  Сформировать список признаков из списка  :param node\_id:  :param features: список признаков, в формате ['Признак 1', 'Признак 2', ...]  :param group\_name: Наименование группы (можно оставить пустым)  :return: словарь признаков для свойств узлов графа на фронте  """  if not group\_name:  group\_name = self.get\_name()  if len(node\_id) > 5:  node\_id = node\_id[-5:]  return dict(map(lambda v: (f'{node\_id}:{v}', {'group': group\_name, 'content': v}), features))  def get\_id(self):  return self.get\_name()  def get\_name(self):  """  [Optional] Returns the \_\_name\_\_ of instance  :return:  """  if self.instance is not None:  return self.instance.\_\_name\_\_  return ''  @staticmethod  def get\_instance\_name(model=None, instance=None, \*\*kwargs):  if instance is None:  return model.\_\_name\_\_  return instance.\_\_name\_\_  @staticmethod  def get\_instance\_id(model=None, instance=None, \*\*kwargs):  return AbstractModule.get\_instance\_name(model, instance, \*\*kwargs)  def validate\_nan\_values(self, df):  is\_nan\_columns = df.isna().any()  assert not is\_nan\_columns.any(), 'The input dataset contains nan values in columns: ' \  f'{is\_nan\_columns[is\_nan\_columns.values].index.tolist()}'  def validate\_target\_selected(self, node, target\_column, \*\*kwargs):  errors, warning = [], []  if target\_column is None:  errors.append('Целевой столбец не выбран (выбирается во входящих ребрах)')  return errors, warning  def validate(self, node, target\_column, \*\*kwargs):  errors, warning = [], []  return errors, warning  @property  def is\_plotting(self):  """  if to plot the data of this model  :return:  """  return len(self.plots\_model) > 0 or len(self.plots) > 0  def get\_columns(self, node):  return getattr(self, 'columns', [])  def get\_plots(self, is\_model=False, \*\*kwargs):  plots = self.plots  if is\_model:  plots = dict(\*\*self.plots\_model, \*\*plots)  return dict(map(lambda kv: (kv[0], kv[1].get\_template(plot\_type=kv[0], \*\*kwargs)), plots.items()))  def plot(self, df, target\_column, additional\_data):  assert 'plot\_type' in additional\_data, 'argument "plot\_type" must be in arguments'  plot\_type = additional\_data.get('plot\_type')  plotting = self.plots[plot\_type] if plot\_type in self.plots else self.plots\_model[plot\_type]  data = df['df'] if isinstance(df, dict) else df  plots = {  plot\_type: plotting.plot(data, target\_column, \*\*additional\_data)  }  return plots  @abstractmethod  @process\_arguments  def call(self, df, target\_column, model\_parameters, \*\*kwargs):  """  The calling of the node  :param df: translated DataFrame collected from parent nodes.  :param target\_column:  :param kwargs:  :return:  """  pass  @abstractmethod  def apply(self, df):  """  Применить обученную модель на передаваемых данных df. Исспользуется только в тех моделях, где есть обучаемая  модель. Например RegressionModels.  :param df:  :return: Результаты применения. Обычно это словарь, например {"predict": [...]}  """  pass  def get\_template(self) -> str:  """  Сформировать html код с отображением результатов (прототип для наследуемых классов). Пример RegressionModels  :return:  """  return ''  def build\_table\_by\_df(self, df, label=''):  template = Template("""{% load i18n %}  <div class="row">  <div class="col-12">  {% if label %}  <label>{% trans label %}:</label>  {% endif %}  <table class="table datatable">  <thead>  <tr>  {% for col in columns %}  <th>{% trans col %}</th>  {% endfor %}  </tr>  </thead>  <tbody>  {% for row in data %}  <tr>  {% for val in row %}  <td>{{ val }}</td>  {% endfor %}  </tr>  {% endfor %}  </tbody>  </table>  </div>  </div>""")  context = Context({  'columns': df.columns.tolist(),  'data': df.values,  'label': label  })  return template.render(context)  def copy(self):  return self  # TODO Сделать нормальный метод copy  # if hasattr(self.instance, 'copy'):  # return self.instance.copy()  # else:  # return self.instance  def update\_features(self, node, features: dict):  """  [Optional] To manipulate the input features from features nodes  :param node: model node  :param features: dictionary of features (Format - see "get\_extra\_features" method)  :return:  """  return features |

* 1. **Сведения о выходных данных**

Результаты работы могут визуализироваться посредством:

* Таблиц (рис. 4.2.1).
* Графиков различного типа (рис. 4.2.2 – 4.2.10).

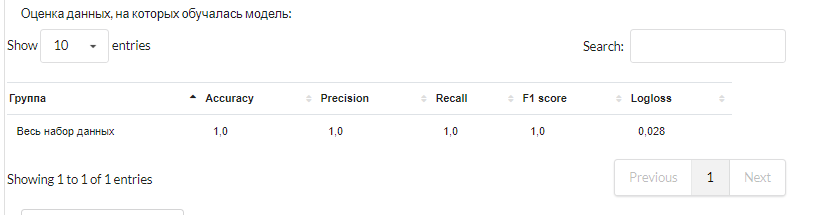


Рисунок 4.2.1 – Таблица оценки данных, на которых обучалась модели логистической регрессии



Рисунок 4.2.2 – Гистограмма влияния признаков на результаты моделирования проекта «Классификация»

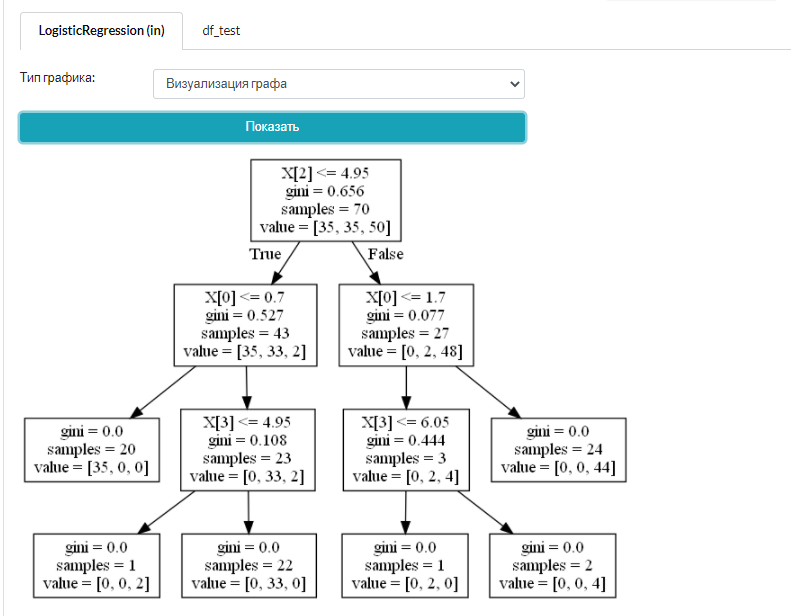


Рисунок 4.2.3 – Дерево графа результатов моделирования проекта «Классификация»

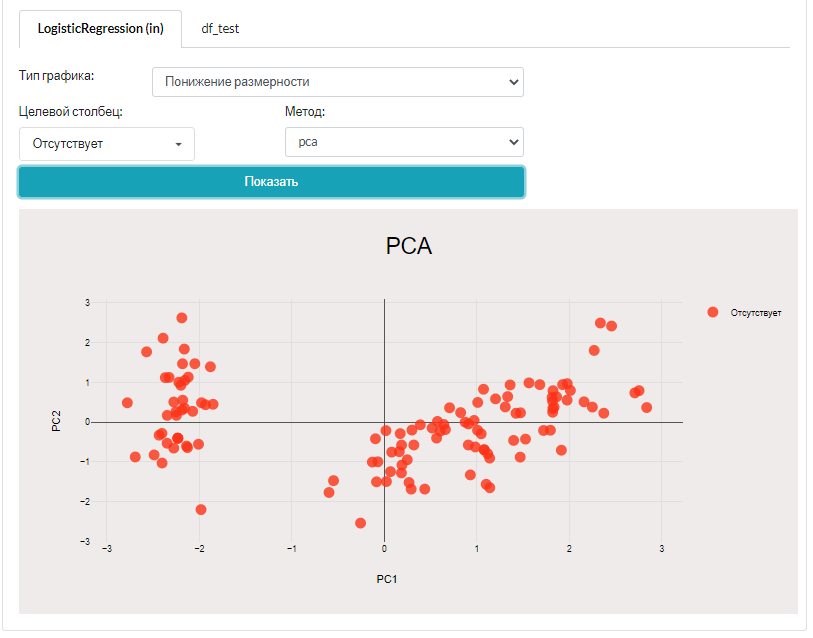


Рисунок 4.2.4 – Точечный график понижения размерности при моделировании проекта «Классификация»

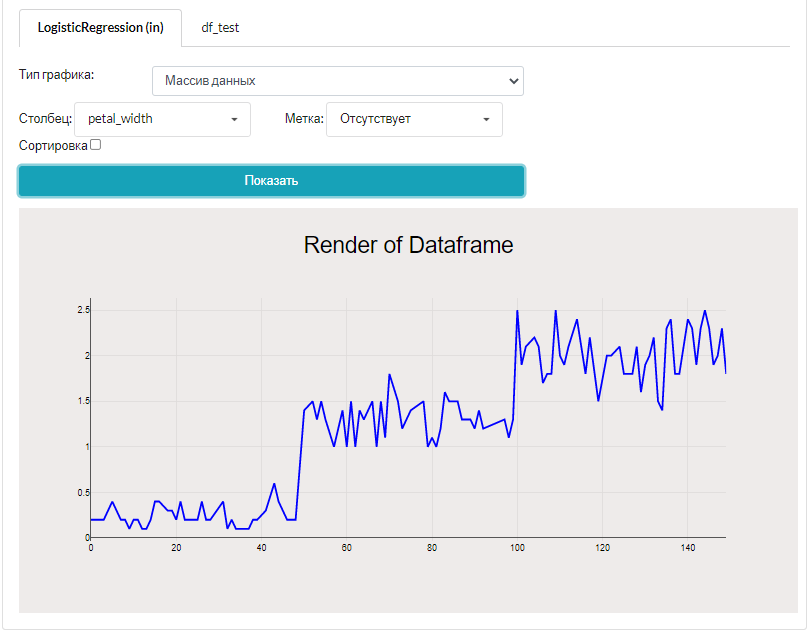


Рисунок 4.2.5 – Линейный график массива данных, полученных при моделировании проекта «Классификация»



Рисунок 4.2.6 – Диаграмма рассеивания при моделировании проекта «Классификация»

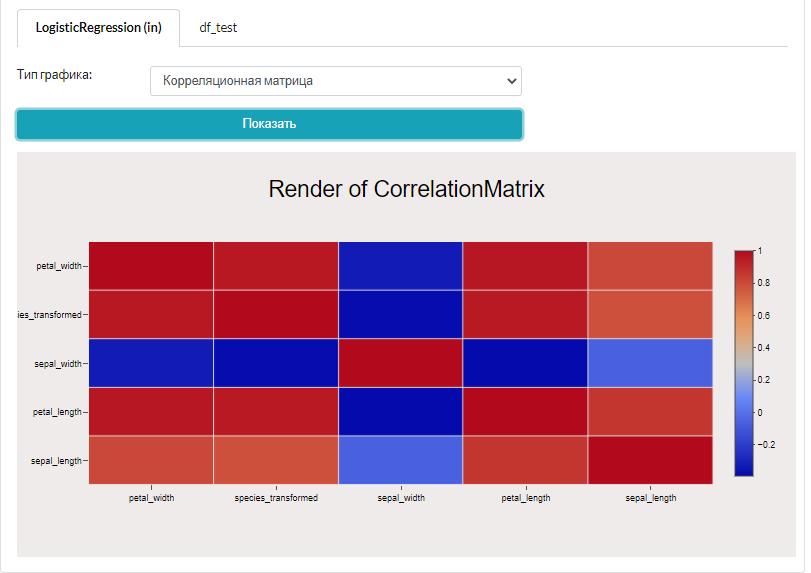


Рисунок 4.2.7 – Корреляционная матрица при моделировании проекта «Классификация»

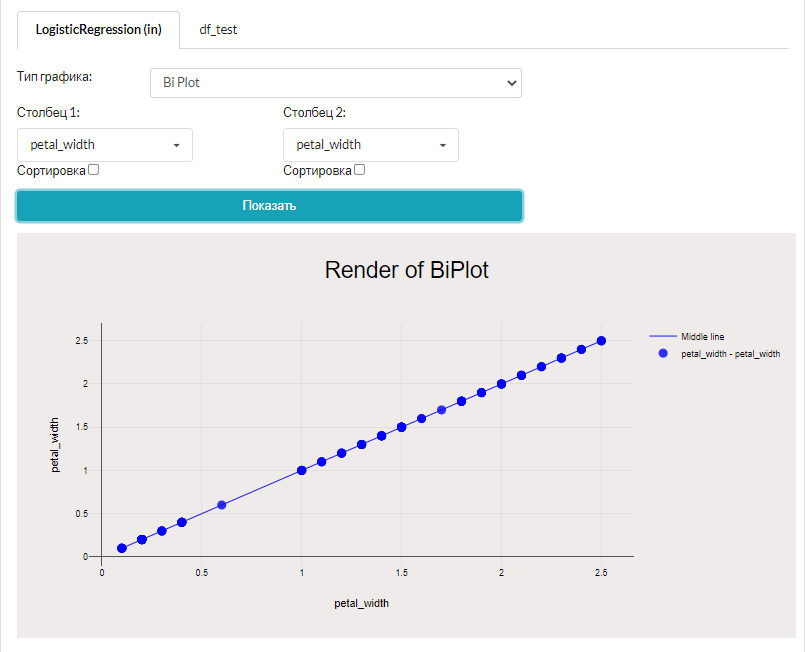


Рисунок 4.2.8 – Bi plot моделирования проекта «Классификация»

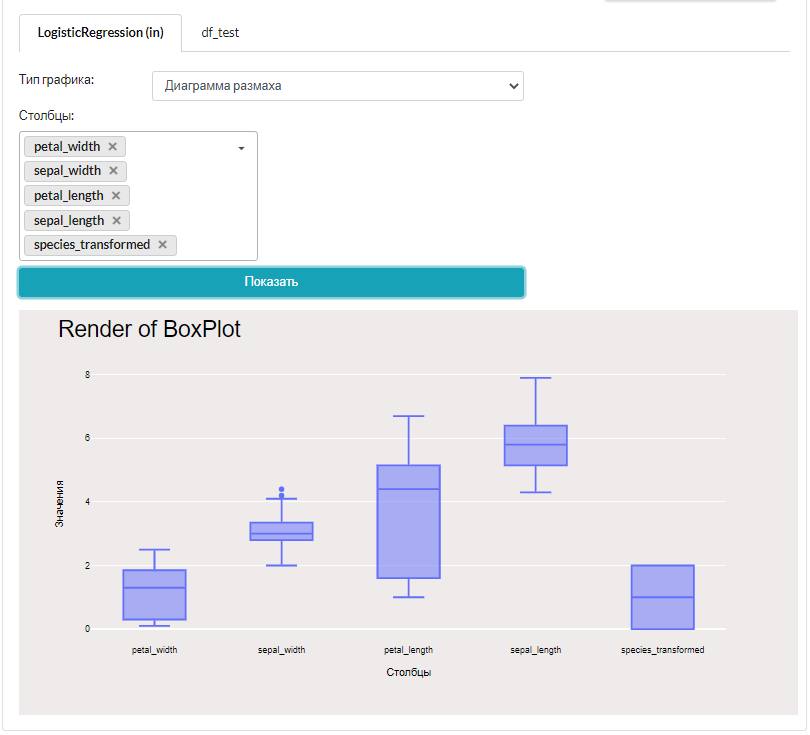


Рисунок 4.2.9 – Диаграмм размаха при моделировании проекта «Классификация»

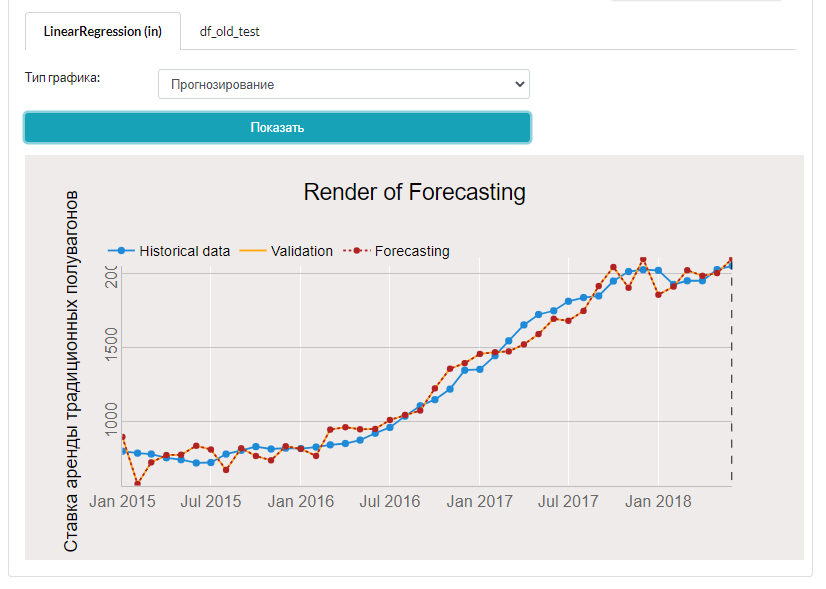


Рисунок 4.2.10 – График результатов моделирования проекта «Ценообразование вагонов»

В ходе проектирования можно скачать в формате PDF-файла отчеты с каждого раздела:

* Отчет для подраздела «Данные» содержит перечень задействованных в проекте файлов с данными, используемыми для модели (рис. 4.2.11).
* Отчет для подраздела «Редактор» содержит сетевой граф проекта (рис. 4.2.12).
* Отчет для подраздела «Анализ данных» содержит таблицы с данными, используемыми при работе модели (рис. 4.2.13).
* Отчет для подраздела «Вычисления» содержит результаты работы модели проекта (рис. 4.2.14).

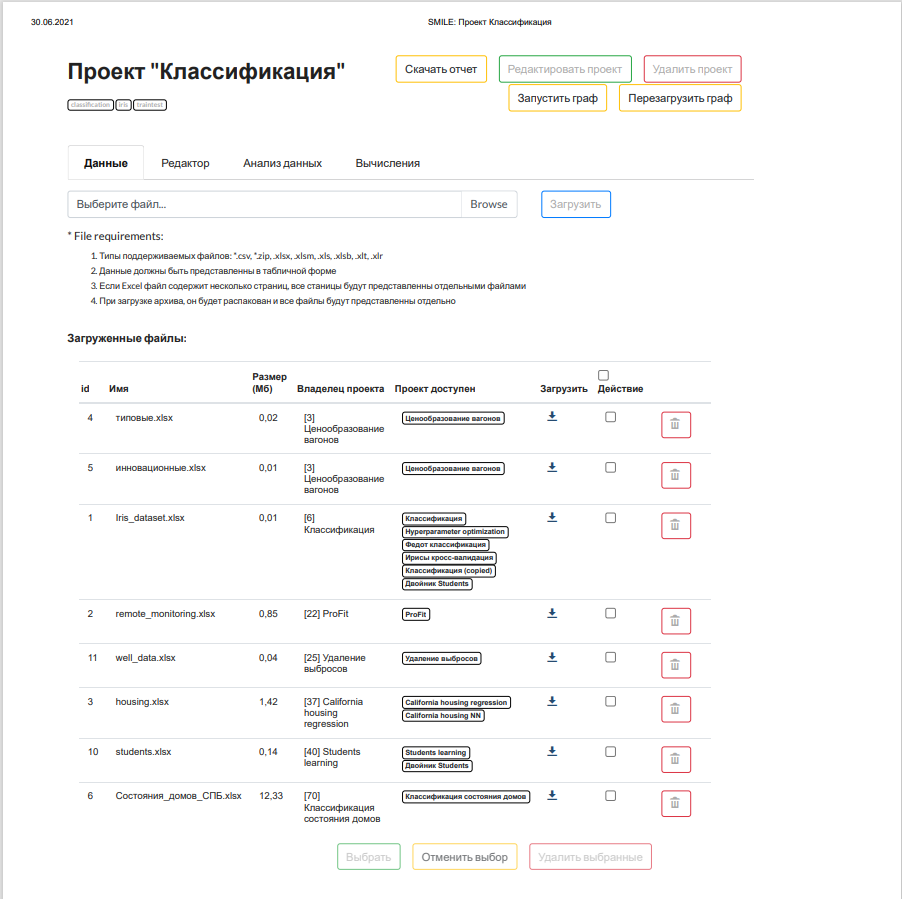


Рисунок 4.2.11 – Отчет для подраздела «Данные» проекта «Классификация»

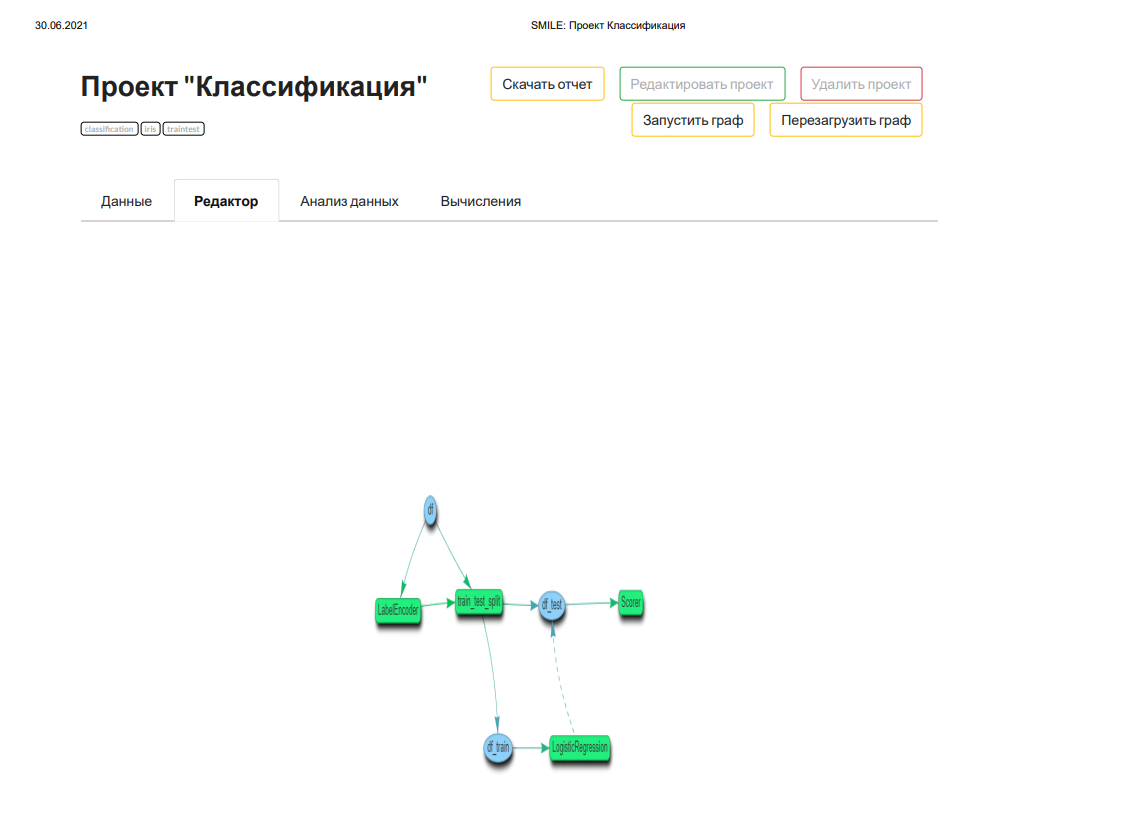


Рисунок 4.2.12 – Отчет для подраздела «Редактор» проекта «Классификация»

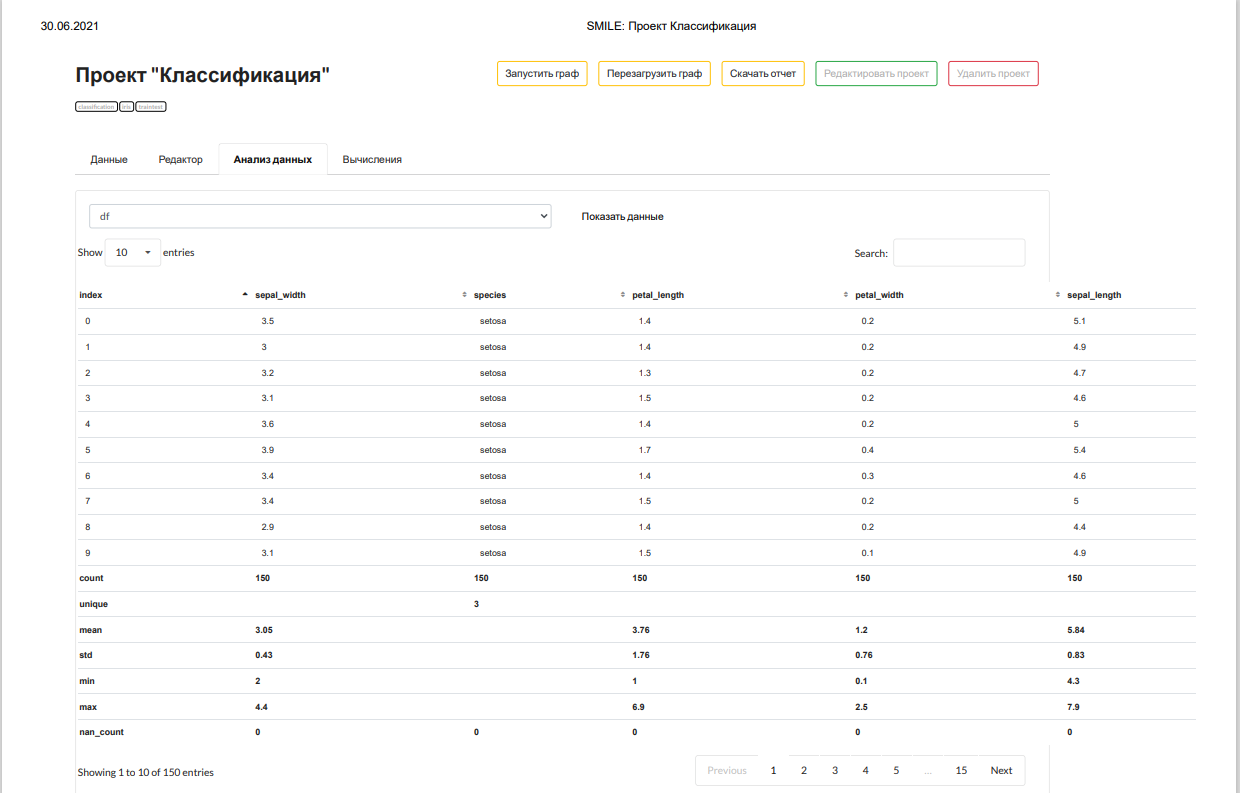


Рисунок 4.2.13 – Отчет для подраздела «Анализ данных» проекта «Классификация»

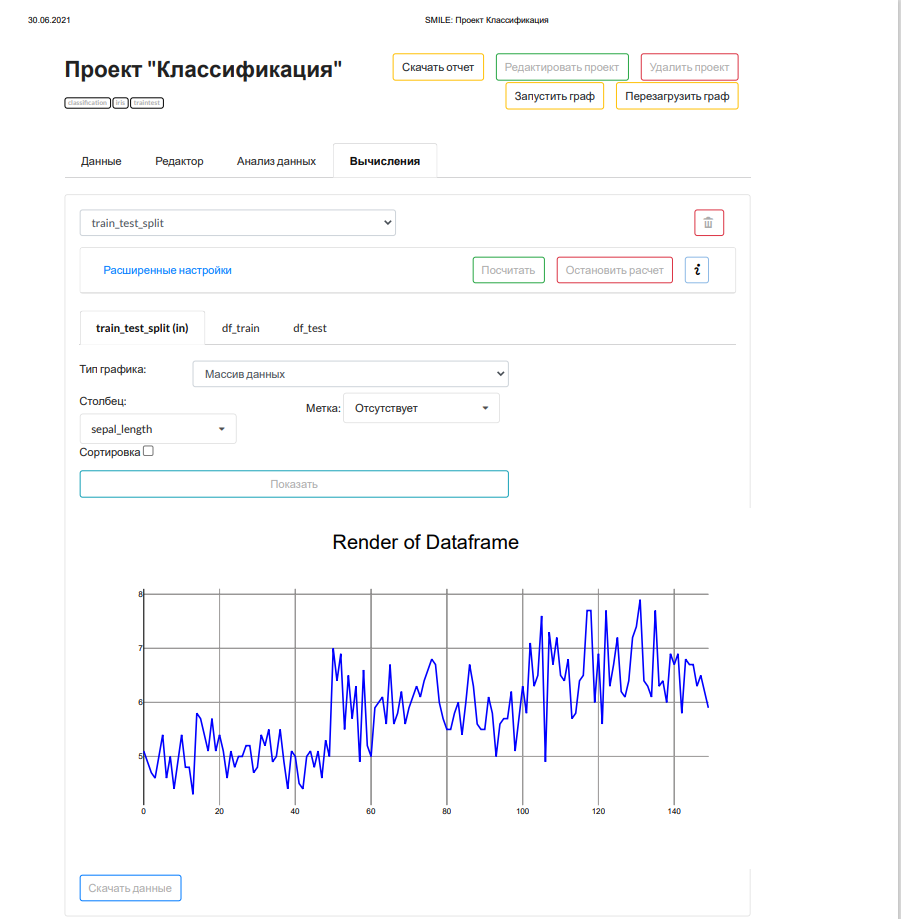


Рисунок 4.2.14 – Отчет для подраздела «Вычисления» проекта «Классификация»

При нажатии в разделе «Вычисления» редактора проектов на кнопку «Скачать данные» происходит загрузка файлов Excel с таблицами данных, полученных в результате моделирования (рис. 4.2.15).

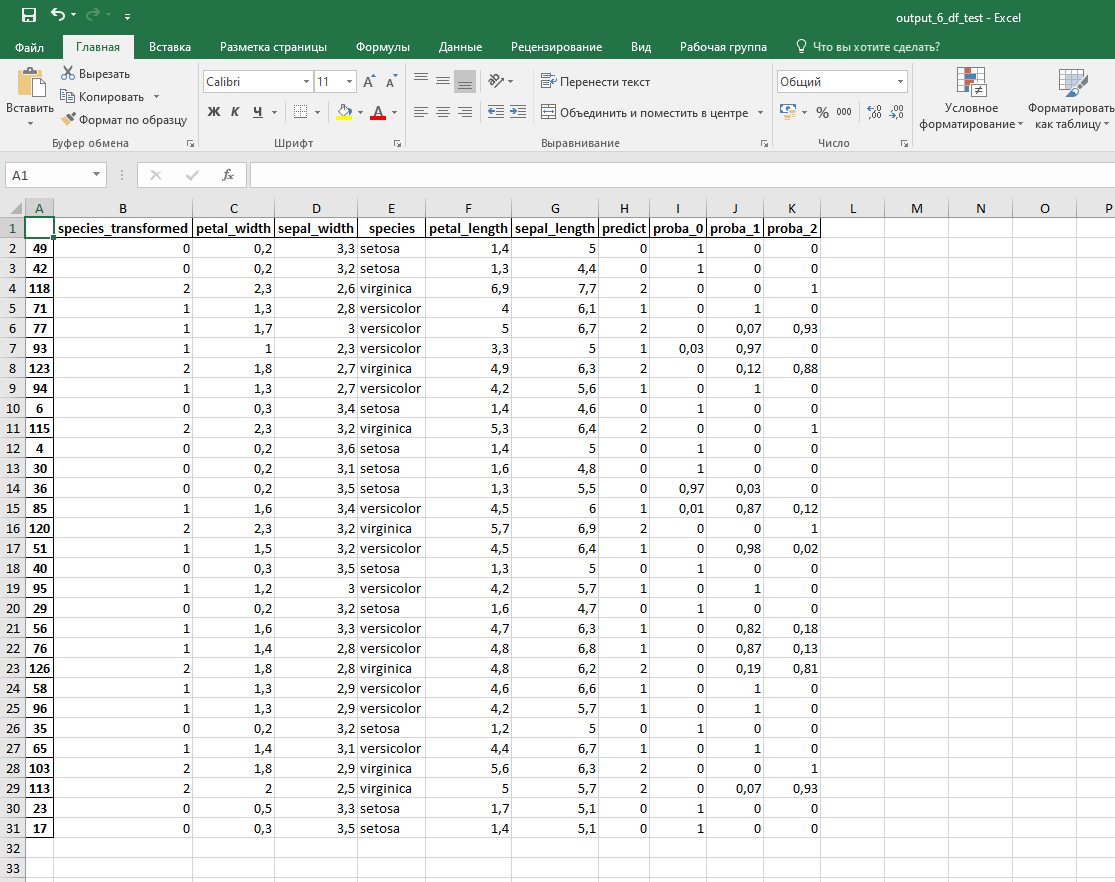


Рисунок 4.2.15 – Файл Excel с результатами работы проекта «Классификация»