МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программной и системной инженерии

РЕКОМЕНДОВАНО К ЗАЩИТЕ В ГЭК
Заведующий кафедрой
Д.т.н., профессор
_____ *А.Г. Ивашко*
_____2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
магистерская диссертация

РАЗРАБОТКА СИСТЕМЫ ПРОГНОЗИРОВАНИЯ УЧАСТНИКОВ ГОСУДАРСТВЕННЫХ ЗАКУПОК ПО ФЕДЕРАЛЬНЫМ ЗАКОНАМ №44-ФЗ И №223-ФЗ

09.04.03 Прикладная информатика
Магистерская программа «Информационные системы анализа данных»

Выполнил (а) работу
Студент (ка) 2 курса
очной формы обучения
Чернушенко
Дарья
Александровна

Научный руководитель
к.т.н, доцент
Цыганова
Мария
Сергеевна

Рецензент
к.ф.-м.н, доцент
Семихин
Дмитрий
Витальевич

г. Тюмень, 2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Количество электронных торгов растет ежегодно, соответственно растет и количество тендеров, которые необходимо проанализировать на предмет вероятности участия и победы. Прогнозирование участников и победителей позволит снизить затраты в закупочной деятельности и сфокусироваться на действительно потенциальных тендерах. В ближайшем будущем есть возможность полностью автоматизировать процессы поиска, анализа вероятности победы (выбора перспективных тендеров), автоматизации сбора и подачи документов и определения участников, победителей. Прогнозирование в части участия и победы является важным и необходимым звеном в данной деятельности. С другой стороны, прогнозирование может позволить заказчикам лучше понять конкурентную среду размещаемых заказов и более точно и экономно планировать бюджеты, выделяемые на закупочную деятельность.

Анализ государственных закупок весьма полезен для определения стратегии предприятия, направленной на повышение его экономических показателей. Например, наличие на соответствующем рынке значительного количества закупок при отсутствии высокой конкуренции дает веские основания изучить вопрос выхода на данное направление. И наоборот, в случае, когда рынок перегружен предложениями, а средняя стоимость снижения цены контракта очень низка, получить маржинальный доход не представляется возможным. В этой ситуации решение об участии в закупках может приниматься только с целью получения положительного имиджа предприятия при условии жесткого контроля затрат.

Для успешной подготовки и защиты выпускной квалификационной работы автором ВКР использовались средства и методы физической культуры и спорта с целью поддержания должного уровня физической подготовленности, обеспечивающую высокую умственную и физической работоспособность. В режим рабочего дня включались различные формы организации занятий физической культурой (физкультпаузы,

физкультминутки) с целью профилактики утомления, появления хронических заболеваний и нормализации деятельности различных систем организма.

В рамках подготовки к защите выпускной квалификационной работы автором созданы и поддерживались безопасные условия жизнедеятельности, учитывающие возможность возникновении чрезвычайных ситуаций.

Глава 1. Постановка цели и задач

Для поиска и выбора выгодных контрактов существует система Метатендер. Метатендер – облачная система, предназначенная для поиска и выбора выгодных или важных контрактов для поставщика по требованиям 44 и 223 Федеральных законов или коммерческих электронных торговых площадок в электронном виде [1]. Цель информационной системы Метатендер – увеличение количества побед авторизированных пользователей в электронных торгах в два раза. С помощью Метандера можно найти выгодный тендер, узнать все о закупках в своем регионе, заключить контракт по желаемой закупке. Данная система является разработкой компании ООО «Электронный Эксперт».

В системе реализованы следующие модули:

- внешнее веб-приложение системы;
- система управления доступами и лицензиями;
- система поиска тендеров;
- система отражения карточки конкретной закупки;
- система сбора избранных тендеров;
- систему совместной работы с избранными тендерами;
- система сопровождения клиентов по выбранным ими закупкам;
- система уведомлений;
- система аналитики и принятия решений.

В системе нет функционала прогнозирования участников закупок, однако это важная составляющая закупочной деятельности. Таким образом, актуальность настоящей работы определяется необходимостью в создании единой системы прогнозирования участников для помощи в принятии решений в закупочной деятельности.

В результате прогнозирования ожидается оценка вероятности участия конкретного поставщика (на основании предыдущих участий).

В рамках данной работы входными данными является информация о закупках по 44 федеральному закону (“О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд”). В закупках по 44-ФЗ заказчиками являются государственные и муниципальные бюджетные учреждения, а поставщиками – любое юридическое или физическое лицо, которые не имеет задолженности по налогам, судимости и не принадлежит к оффшорным компаниям.

Цель работы: повышение эффективности закупочной деятельности пользователей системы "Метатендер" путем разработки и интеграции в систему модуля прогнозирования участников государственных закупок по 44 Федеральному закону.

В качестве показателей эффективности закупочной деятельности рассматривается вероятность участия и победы поставщика, что позволит снизить затраты и сфокусировать внимание на потенциальных тендерах.

Для достижения поставленной цели необходимо решение следующих **задач**:

1. Изучение процессов организации и сопровождения электронных торгов.
2. Изучение архитектуры системы Метатендер.
3. Извлечение и первичный анализ данных о закупках.
4. Проектирование и реализация процедур очистки первичных данных.
5. Конструирование входных признаков для модели прогнозирования.
6. Исследование значимости сформированных признаков.
7. Построение модели прогнозирования участников закупок.
8. Тестирование модели.
9. Изучение возможных методов интеграции модуля прогнозирования и системы Метатендер.

Глава 2. Предварительная обработка данных

2.1 Архитектура системы Метатендер

Метатендер – облачная система, предназначенная для поиска и выбора выгодных или важных контрактов для поставщика.

Архитектура системы Метатендер (рисунок 1):

1. Web Server – веб-приложение сайта [1].

   Стэк: Сейчас работают следующие приложение: MetaTender (Net Framework 4.6.1), MetaTender.API (Net Framework 4.6.1), MetaTender.UserCrm (Net Core).

   Конфигурация: Windows Server 2016 Datacenter, VCPU x 2, RAM 4GB, HDD 60 GB.

   Приложения: IIS 7.5+, Net Framework 4.6.1, MySQL Net Connector, Net Core Runtime 2.1, NET Core Windows Server Hosting

2. Сервер базы данных – осуществляет хранение данных веб приложения и обработанных (отпарсеных) данных закупочных процедур (все данные).

   Конфигурация: CentOS 7, VCPU x 4, RAM 32GB, HDD 220 GB, SSD 1100 GB.

   Приложения: MariaDB 5.5.64

3. Сервер поиска – осуществляет хранение данных поиска для быстрого поиска по закупочным процедурам по которым осуществляется поиск.

   Конфигурация: CentOS 7, VCPU x 4, RAM 32GB, HDD 100 GB, SSD 100 GB.

   Приложения: SphinxSearch 2.2.11

4. Сервер событий (оперативные данные) – предназначен для работы с кэшем (для скорости) (Redis) и очередью сообщений (хранит очередь и дает возможность их обрабатывать) (RabbitMQ).

   Конфигурация: CentOS 7, VCPU x 2 (20%), RAM 4GB, HDD 50GB

Приложения: RabbitMQ, Redis

5. Сервер микро-сервисов - на данном сервере запущены все сервисы:

- collector – сервис по сбору и парсингу закупок с разных источников;
- events – сервис по обработке событий (рассылка сообщений клиентам и управление тикетами);
- general – сервис по обработке системных событий и отображения статистики (посещаемость, ошибки, статистика);
- searchResource – сервис по записи и обработке поисковых событий, обновление шаблонов клиентских поиска;
- winners – сервиса поиска победителей на zakupki.gov.ru.
- tekTorg – сервис парсинга закупочных процедур ЭТП ТэкТорг.
- zakupki_mos_ru – сервис парсинга закупочных процедур Портала Поставщиков.
- sphinx – новый поиск тендеров.

Конфигурация: Centos, VCPU x 4 (50%), RAM 8 GB, HDD 100GB

Приложения: Docker Services Server

6. Почтовый сервер – рассылка, получение и хранение электронной почты.

Конфигурация: CentOS 7, VCPU x 1, RAM 1GB, HDD 60GB

Приложения: Postfix, Dovecot, Php 5.6, Nginx, Certbot

7. Сервис: объектное хранилище - для распределенного хранения документов с разных сервисов. Арендуется у Yandex.Cloud.

8. Сервис прокси – необходимо для некоторых парсеров площадок. Арендуются у провайдера Proxy4.net

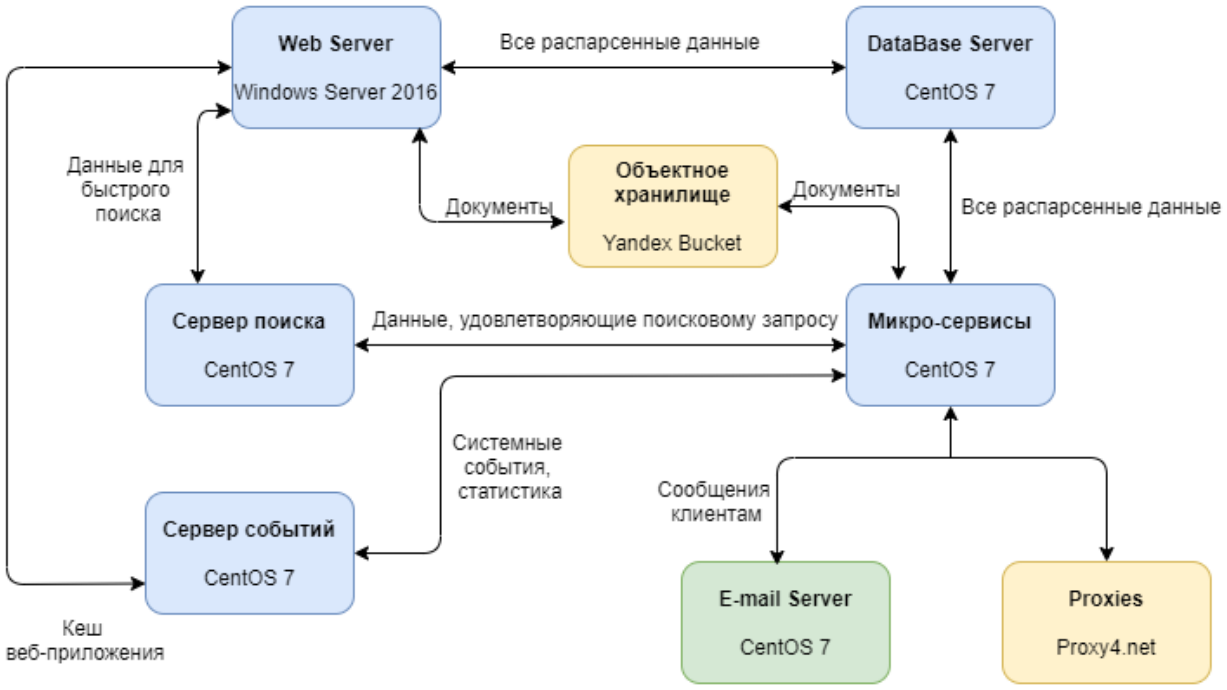

Рисунок 1 – Архитектура системы Метатендер

На рисунке 1 голубыми блоками выделены сервера на Яндекс облаке, желтыми – сервисы, зеленым – виртуальный сервер на собственных ресурсах.

2.2 Извлечение данных из первичного источника

В качестве источника данных, подлежащих анализу, использовалась MySQL база данных, сформированная специалистами компании ООО «Электронный Эксперт» в результате парсинга официального сайта единой информационной системы в сфере закупок [2]. Данные из этой базы данных также используются в системе Метатендер. База данных содержит 25 таблиц с различной информацией, однако информация о закупках находится в 3 таблицах: contract, purchase, protocol_app. Напрямую из модуля прогнозирования подключение к базе данных осуществить не представляется возможным на текущий момент, в виду технических особенностей сервера базы данных, поэтому с помощью SQL-запроса были получены данные, которые будут использованы в дальнейшем. Полученные из базы данных записи были выгружены в csv-файл, который является источником входных

данных. В дальнейшем, когда будут увеличены мощности сервера базы данных, закупочные данные можно будет выгружать напрямую из базы с помощью компонентов Loginom.

Рисунок 2 – Структура базы данных

Таблицы в базе данных связаны между собой по идентификатору закупки – выделен желтым цветом. Из всех данных, представленных в таблицах базы данных, были использованы поля, обозначенные зеленым цветом, поскольку они несут необходимую информацию о закупках.

2.3 Анализ исходных данных

Структура полученного набора данных показана в таблице 1. Период сбора исходных данных: с января 2018 года по апрель 2021 года. Первичный набор данных содержал 4 тысячи 607 записей.

Таблица 1 – Структура исходного набора данных

Описание поля	**Наименование**	**Формат**
Реестровый номер извещения (закупки, для нас - идентификатор)	purchaseNumber	Строковый
Дата опубликования закупки	publicationDateTime	Дата/время
Закон закупки (44)	purchaseType	Целый
Заказчик (ИНН как идентификатор)	customerInn	Строковый
Цена тендера	price_tender	Вещественный
Наименование закупки	title	Строковый
Номер лота закупки	lot_num	Целый
Классификатор закупки по ОКВЭД	okved	Строковый
НМЦК лота	nmck	Вещественный
Итоговая цена лота	price_lot	Вещественный
Итоговая цена контракта	price_contract	Вещественный
Победитель	winner	Логический
Участник лота (ИНН как идентификатор)	providerInn	Строковый
Дата опубликования контракта	contacDateTime	Дата/время

Из данных, приведенных в таблице 1 не все признаки будут являться входными, поскольку для прогнозирования не важны значения:

- реестрового номера извещения, так как оно уникально для каждой закупки;
- номер закона закупки также не будем учитывать, так для анализа выбирались данные по 44 ФЗ;
- ИНН заказчика и поставщика, потому что они обозначают только идентификатор участников;
- номер лота закупки, так как он обозначает количество лотов.

Поэтому, первичными входными признаками для модели прогнозирования будут являться:

- название закупки;
- ОКВЭД;
- дата публикации закупки;
- цена тендера;
- НМЦК лота;
- дата публикации контракта.

Полученных признаков недостаточно для дальнейшего прогнозирования, ввиду чего необходимо сформировать другие входные признаки на основании текущих.

Выходной признак: логическое поле (true, false), которое должно обозначать может ли поставщик являться участником конкретной закупки.

В рамках рассматриваемой задачи подготовка данных включает в себя:

- анализ данных на наличие пропусков, ошибок, дубликатов, противоречий и т. п.;
- реализацию процедур обработки пропусков;
- обработку ошибочных записей;
- удаление дубликатов;
- поиск и удаление противоречий.

Таким образом, основные этапы реализации модуля прогнозирования представлены на рисунке 3.

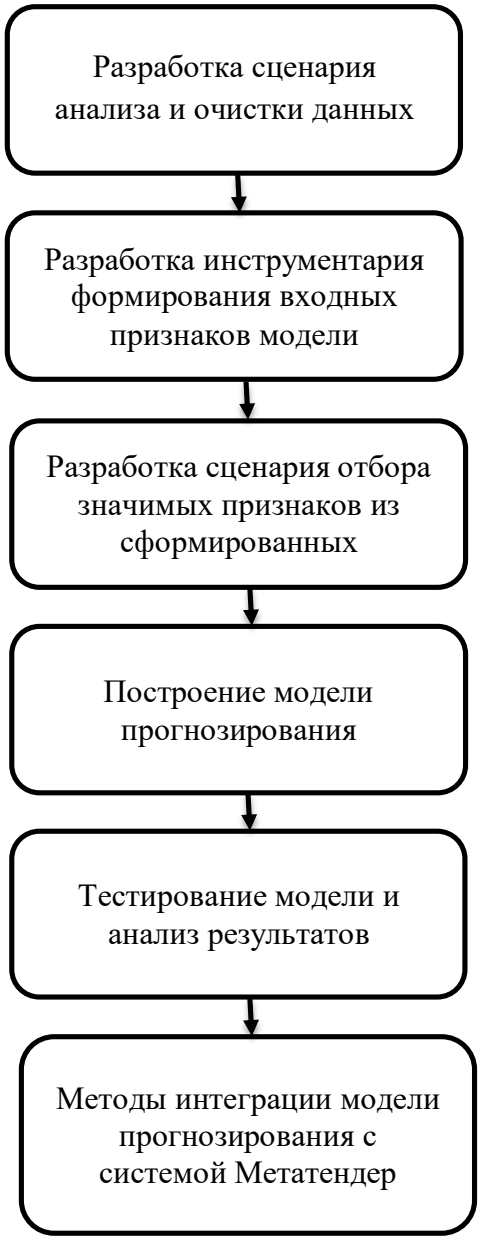

Рисунок 3 – Блок-схема по основным этапам разработки модуля прогнозирования

2.4 Реализация процедур очистки данных

Для организации процедур предварительной обработки данных и получения требуемых показателей была выбрана аналитическая платформа Loginom – low-code-платформа, предоставляющая инструментарий реализации всех аналитических процессов от интеграции и подготовки данных до моделирования, развертывания и визуализации [3].

Часть сценария, в рамках которого выполнялась процедура очистки данных представлена на рисунке 4.

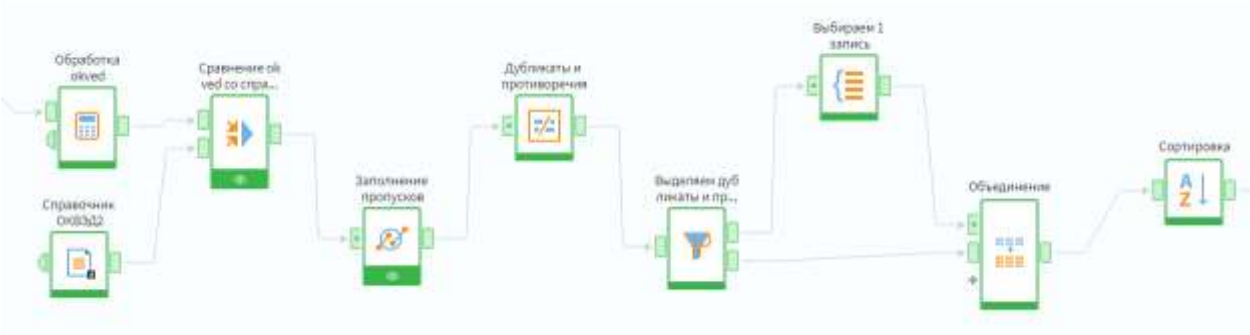

Рисунок 4 – Схема выполнения предварительной обработки данных

Обнаружение пропусков выполнялось с помощью встроенного визуализатора Статистика. Наличие пропущенных значений было выявлено в большинстве столбцов анализируемого набора, следствием чего стала необходимость создания специальных процедур обработки пропусков. Выбор того или иного метода обработки (восстановление, удаление записей с пропусками и т. д.) определяется долей записей с пропусками, а также характером пропусков и возможностью применения каких-либо процедур восстановления в данном конкретном случае. Решение принималось по каждому полю отдельно. В таблице 2 приведены сведения о пропусках и принятые решения относительно метода их обработки.

Таблица 2 – Количество пропусков в исходном наборе данных и метод их обработки

Столбец	**Количество пропусков**	**Метод обработки**
purchaseNumber	0	без изменений
publicationDateTime	0	без изменений
purchaseType	32	удаление записей
customerInn	0	без изменений
price_tender	0 пропусков 1 запись, заполненных 0	удаление записей
title	0	без изменений
lot_num	0	без изменений (1 уникальное значение, незначимое поле)
okved	1042	удаление записей
nmck	913	удаление записей
price_lot	0 пропусков 1 запись, заполненных 0	удаление записей
price_contract	1 пропусков 6 записей, заполненных 0	удаление записей
winner	1	удаление записей
providerInn	1	удаление записей
COL2	1042	удаление записей
contacDateTime	1	без изменений

Отдельного пояснения требуют процедуры обработки значений в столбце okved: обработка поля с классификатором закупки по ОКВЭД выполнялась с

помощью регулярного выражения: из строки исключались буквы и выбирались первые две пары цифр (**.**). Такой способ представления данных в этом поле в дальнейшем будет удобен для определения отрасли закупки.

В остальных столбцах обработка пропущенных значений выполнялась в соответствии с таблицей 2. Пропуски в незначимых полях игнорировались, в остальных случаях удалялись как записи с пропусками, так и записи, в которых НМЦК цена меньше или равны 0. В результате обработки пропусков было удалено 1042 записи.

Исключение дубликатов и противоречий выполнялось с помощью стандартного компонента Дубликаты и противоречия [3]. Дубликатами считались те записи, значения которых одинаковы по всем полям, а противоречиями – если у записей отличаются поля дата публикации закупки, дата публикации контракта, ИНН заказчика, НМЦК, победитель и цена тендера. Таким образом входными полями при поиске дубликатов и противоречий были выбраны: ОКВЭД, номер закупки, ИНН победителя, выходными: НМЦК, ИНН заказчика, дата публикации закупки и контракта, победитель. После чего, из исходного набора с помощью компонента Фильтр были отобраны записи, которые определены как дубликаты или противоречия (значение True в соответствующем столбце) и далее произведено их удаление. Удаление осуществлялось с помощью компонента Группировка, в качестве измерений использовались поля Группа дубликатов и Группа противоречий, все входные поля рассматривались в качестве показателей с агрегацией «первый» (рисунок 5).

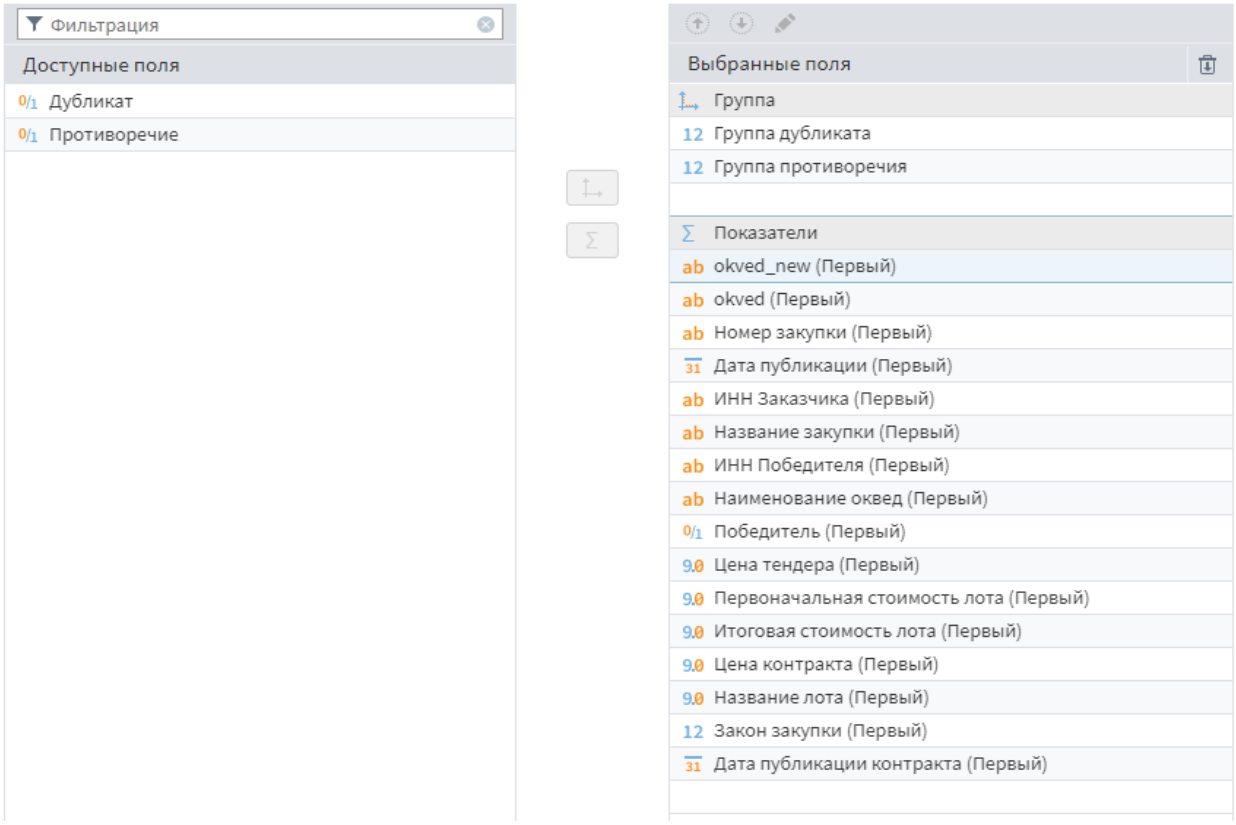

Рисунок 5 – Удаление дубликатов и противоречий

Оставшиеся в результате группировки записи, с помощью стандартного компонента Объединение были добавлены к данным, которые не удовлетворили условиям фильтра (значения полей Дубликат и Противоречие были равны false). Таким образом было удалено 142 дубликата и противоречия.

Глава 3. Формирование входных признаков прогнозирования

Зачастую самым важным при решении задачи является умение правильно отобрать и даже создать признаки (Feature engineering). Feature engineering – процесс использования знаний предметной области для извлечения функций (характеристик, свойств, атрибутов) из необработанных данных [4].

Методы и техники создания признаков [4]:

1) Обработка строковых признаков – обработка признаков, значением которого могут принимать конечное количество строк. Например, возможно определение полового признака.

2) Категориальные признаки – возможна обработка признаков, которые принимают значения на конечном наборе.

Например, столбец цвет, который может принимать значения blue, red, green, или unknown. В таком случае можно добавлять признаки вида is_red, is_blue, is_green, is_red_or_blue и тому подобно.

3) Числовые переменные – округление или разделение на целую и вещественную часть вещественных переменных.

4) Дата и время – добавление признаки, за счет выделения времени дня, количество прошедшего времени с определенного момента, выделение сезонов, времен года, кварталов.

5) Агрегированные признаки – добавление признаков, которые агрегируют другие признаки объекта, сокращая при этом размерность признакового описания (минимальное, максимальное, среднее, количество).

Все входные признаки условно можно разбить на две группы: признаки, характеризующие закупки и признаки, характеризующие участников.

Используя методы и техники создания признаков (выделение сезона, времени года, агрегирование признаков), были сформированы следующие входные признаки, характеризующие закупки:

- количество уникальных участников по всем классификаторам закупок;
- месяц публикации закупки;
- сезон (осень, зима, весна, лето) публикации закупки.

К характеристикам участников можно отнести:

- количество уникальных классификаторов закупки, в которых принимал участие поставщик;
- общее количество побед;
- минимальная и максимальная дата участия у поставщика;
- разница в месяцах между первой и последней датой публикации закупки, в которой принимал участие поставщик;
- минимальная и максимальная стоимость лота у поставщика по каждому классификатору закупки;
- количество участий и побед поставщика по каждому классификатору закупки.

Сформированные признаки могут быть полезны для прогнозирования, однако в первичном наборе входных признаков их нет, поэтому необходимо их вычислить. Общая схема их расчета иллюстрируется сценарием, изображенным на рисунке 6. Далее приведено описание расчета основных показателей. Все признаки должны вычисляться исходя из предшествующих данных [5].

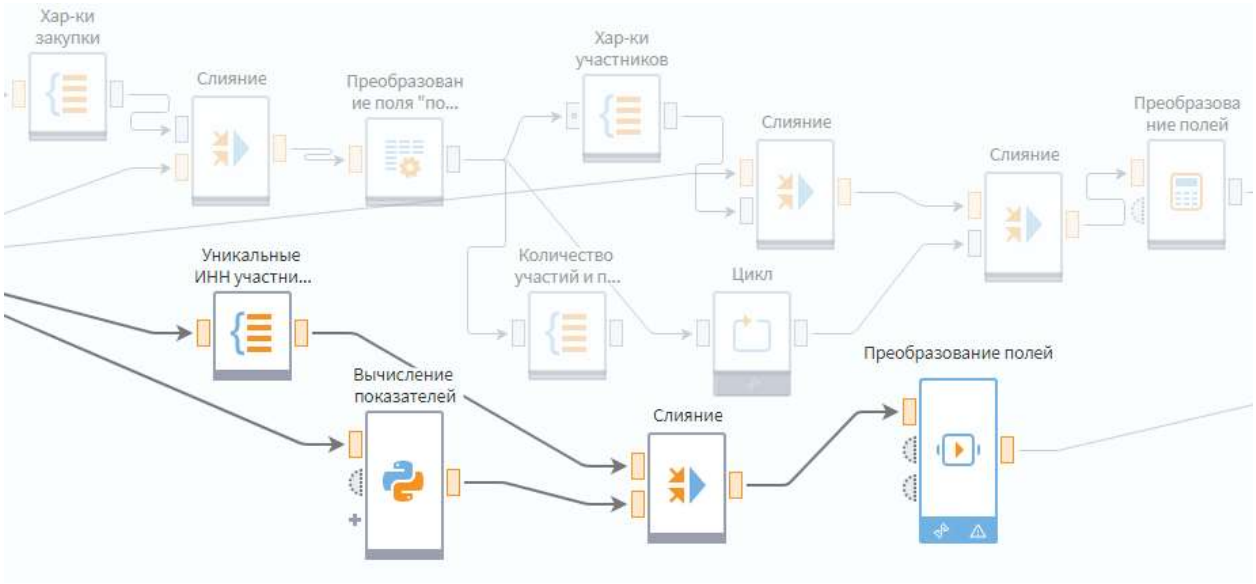

Рисунок 6 – Расчет признаков прогнозирования.

Представленный на рисунке 6 сценарий содержит в себе расчет признаков прогнозирования с использованием платформы Loginom. Основные используемые компоненты: Группировка, Слияние, Цикл с групповой обработкой, Калькулятор для преобразования полей. О каждом используемом компоненте описано далее.

3.1 Характеристики закупки

3.1.1 Количество уникальных участников по всем классификаторам закупок

Для этого использовался стандартный компонент Группировка. Измерение группировки – okved, показатель: ИНН Победителя, функция агрегации: «количество уникальных» (рисунок 7).

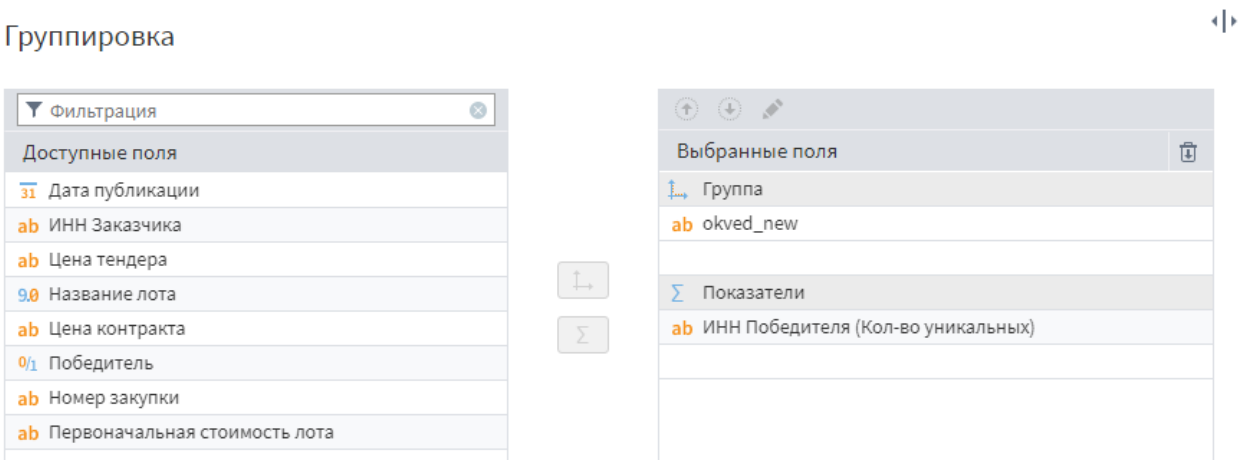

Рисунок 7 – Использование компонента Группировка для вычисления количества уникальных участников по классификаторам закупок.

Далее с помощью компонента Слияние было выполнено добавление полученных в предыдущем пункте данных с исходными данными по полю okved (рисунок 8).

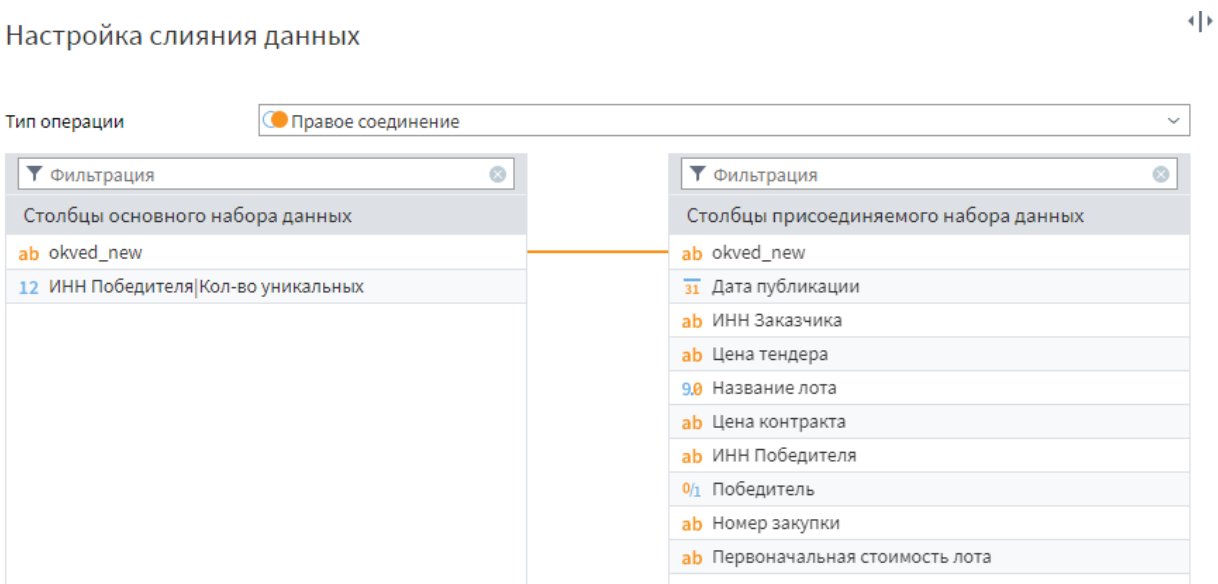

Рисунок 8 – Слияние полученных результатов с обработанными данными.

3.1.2 Месяц публикации закупки

Вычисление выполнялось с помощью стандартного компонента Калькулятор: добавлен столбец month, значение которого вычисляется по формуле 1:

= Month(publicationDateTime), (1)

где publicationDateTime – дата публикации закупки.

3.1.3. Сезон публикации закупки

Значение полей вычислялось с использованием Калькулятора по формуле 2:

$$season = IF(Month(publicationDateTime) <= 2, "зима", IF(Month(publicationDateTime) <= 5, "весна", IF(Month(publicationDateTime) <= 8, "лето", IF(Month(publicationDateTime) <= 11, "осень", "зима")))), \quad (2)$$

где publicationDateTime – дата публикации закупки.

3.2 Характеристики участников

Для формирования характеристик участников был выполнен расчет следующих показателей:

3.2.1 Количество уникальных классификаторов закупки, в которых принимал участие поставщик

На рисунке 9 можно увидеть, что в компоненте Группировка использованы следующие показатели: измерение группировки – ИНН Победителя, показатель: okved, функции агрегации: «количество», «количество уникальных».

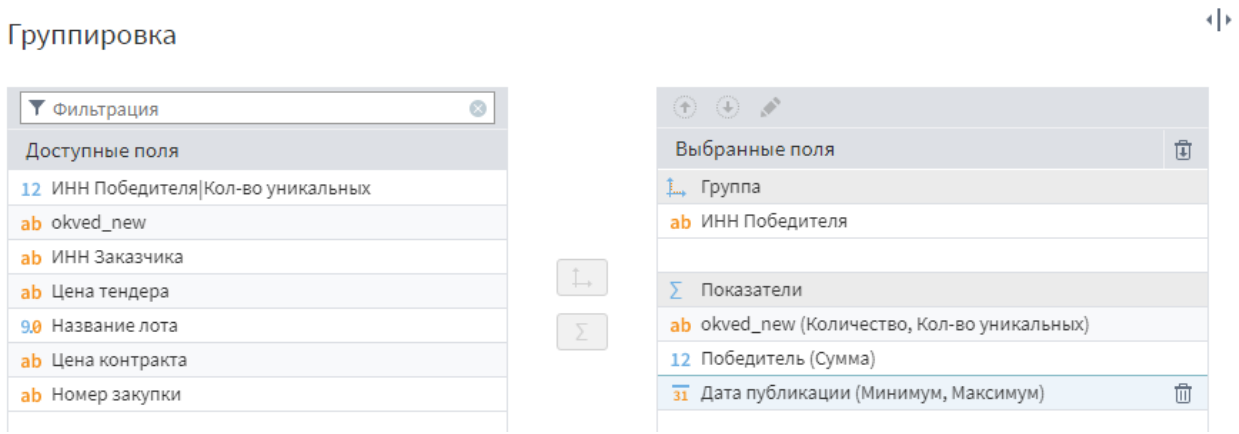

Рисунок 9 – Использование компонента Группировка для вычисления количества уникальных классификаторов закупки, в которых участвовал поставщик.

3.2.2 Общее количество побед

Определено по аналогии с пунктом 3.2.1. С помощью компонента Группировка. Измерение группировки – ИНН Победителя, показатель: победитель (предварительно с помощью компонента Параметры полей, формат поля был преобразован из Boolean в Целый), функция агрегации: «сумма».

3.2.3 Минимальная и максимальная дата участия у поставщика, разница между первым и последним участием

Сначала для каждого поставщика с помощью стандартного компонента Группировка были определены минимальное и максимальное значение даты публикации закупки, в которой он принимал участие.

После чего в компоненте Калькулятор был добавлен столбец time_between, значения которого вычислены по формуле 3:

$$time\_\text{between} = MonthsBetween(publicationDateTime\_Min, publicationDateTime\_Max), \quad (3)$$

где publicationDateTime_Min – первая дата участия поставщика, publicationDateTime_Max – последняя дата участия поставщика.

3.2.4 Минимальная и максимальная стоимость лота у поставщика по каждому классификатору закупки

Значения вычислялись по аналогии с пунктом 3.2.3.

3.2.5 Количество участий и побед поставщика по каждому классификатору закупки

Для вычисления данных полей необходимы уникальные значения okved. Измерение Группировки – ИНН Победителя, okved, показатели: okved, победитель функции агрегации: «количество», «сумма» соответственно (рисунок 10).

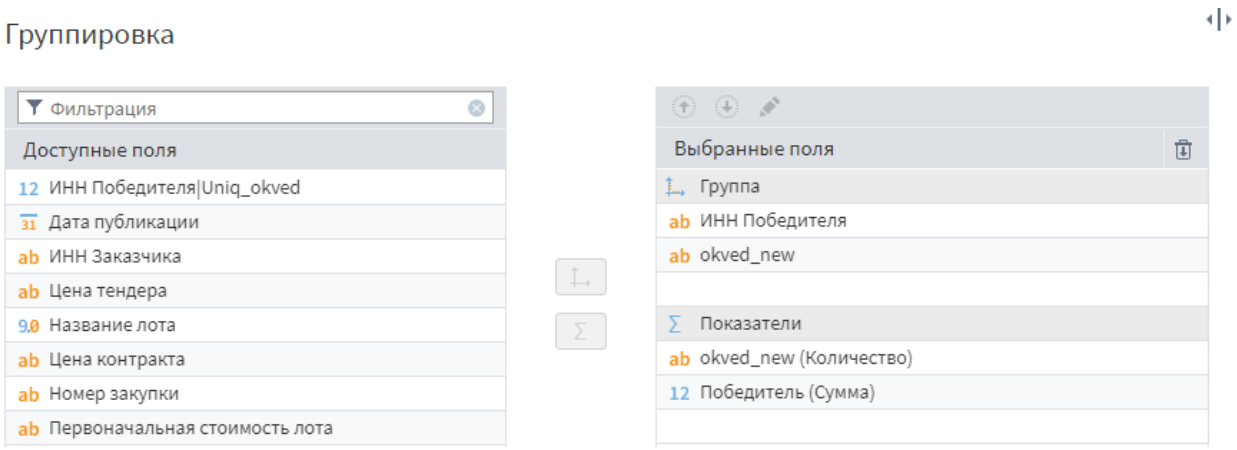

Рисунок 10 – Использование компонента Группировка для вычисления количества участий и побед поставщика по каждому классификатору закупки.

Далее с помощью компонента Цикл, с использованием групповой обработки был выполнен расчет количество участий и побед поставщика по каждому классификатору закупки (рисунок 11). На рисунке 12 представлен результат вычисления количества участий и побед поставщика по каждому классификатору закупки.

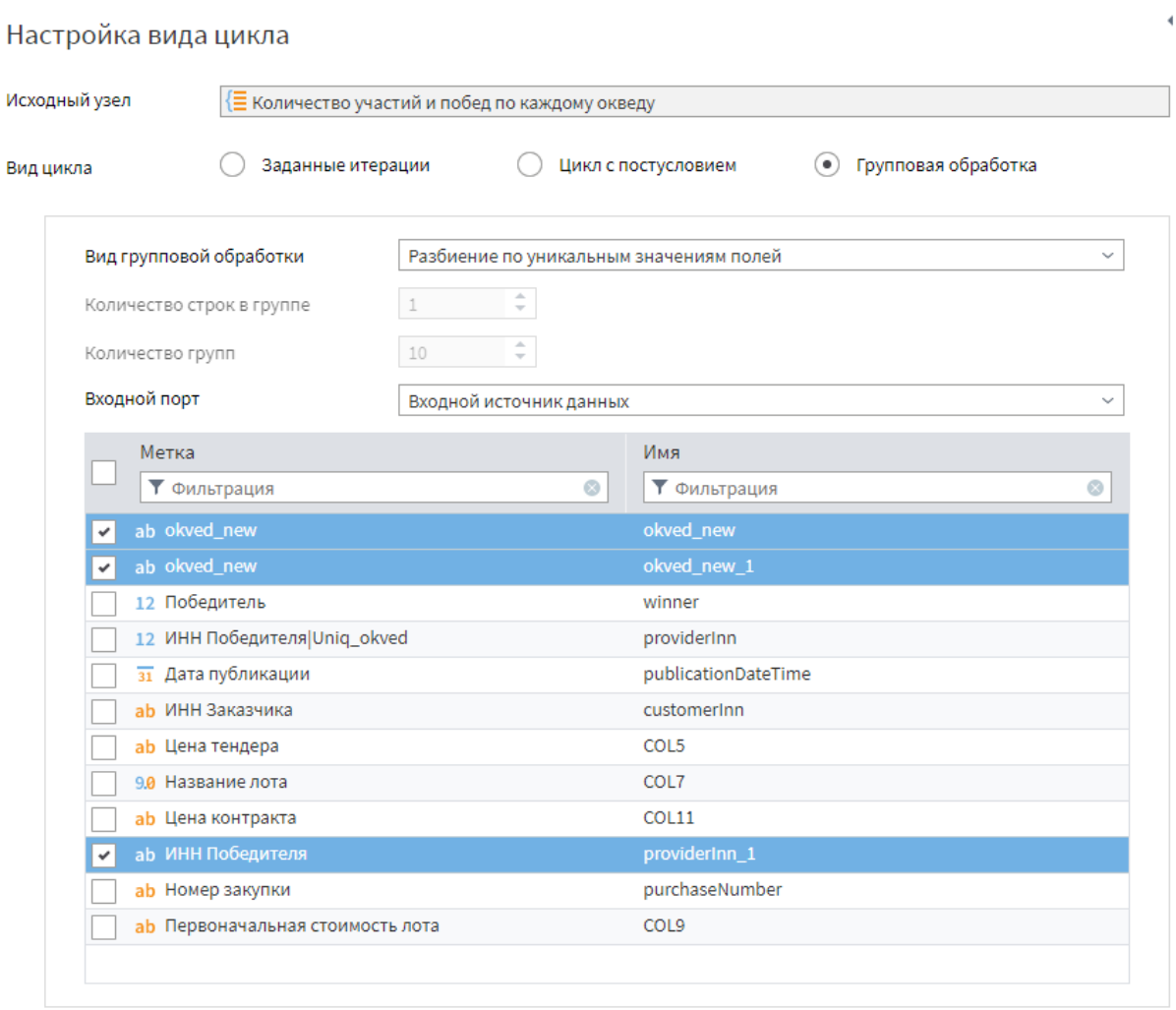

Рисунок 11 – Настройка цикла групповой обработки для вычисления количества участий и побед поставщика по каждому классификатору закупки.

#	ab ИНН Победителя	ab okve...	12 okved_new|Количество	9.0 Победитель|Сумма
1	7204179398	01.13	1	1,00
2	722200475952	01.13	1	1,00
3	7203489478	01.19	1	0,00
4	721700042401	01.19	1	1,00
5	7220100204	01.19	1	1,00
6	720315948810	02.20	1	0,00
7	7206055278	02.20	1	1,00
8	7205024238	08.12	1	1,00
9	7203355844	10.12	1	0,00
10	7204179398	10.12	1	0,00

Рисунок 12 – Результат вычисления количества участий и побед поставщика по каждому классификатору закупки.

После чего выполнено слияние полученных данных (в пунктах 3.1 и 3.2) с исходными данными.

Для корректного прогнозирования данных все признаки должны формироваться из предшествующих данных, то есть при их вычислении необходима фильтрация по дате публикации закупки. В процессе вычисления большинства признаков используется цикл с групповой обработкой, с помощью компонентов Loginom нет возможности динамически фильтровать записи в цикле, поэтому было принято решение вычислять количественные признаки с помощью Python-скрипта (Приложение А).

3.3 Формирование входного набора данных

Входными признаками являются следующие поля (на основании исходной модели и результатов преобразований из пунктов 3.1-3.2):

- номер (purchaseNumber) и название закупки (title);
- оквед (okved);
- преобразованный к маске “**.**”оквед (okved_new);
- дата публикации закупки (publicationDateTime);
- ИНН участника (providerInn);
- ИНН заказчика (customerInn);
- цена тендера (price_tender);
- количество уникальных провайдеров по окведам (count_uniq_prov_on_okved);
- количество всех участий поставщика по окведам (count_part_prov);
- количество уникальных окведов, в которых учавствовал поставщик (count_uniq_okv_on_prov);
- количество побед (count_win);
- количество побед по окведам (count_win_on_okved);
- минимальная дата участия у провайдера (min_date_prov);

- максимальная дата участия у провайдера (max_date_prov);
- минимальная стоимость лота по окведам (min_price_on_okved);
- максимальная стоимость лота по окведам (max_price_on_okved).

Выходной столбец part должен содержать информацию об участии каждого поставщика во всех закупках (true - если участвовал, false - если не участвовал). То есть необходимо выполнить полное слияние всех уникальных ИНН участников с каждой записью исходных данных. Уникальные ИНН участников были определены с использованием Группировки, где измерением являлось поле ИНН участника. Затем выполнено полное слияние данных.

Поле для прогнозирования part, как ранее упоминалось, должно быть логическим. Его значение равно true, если добавленный ИНН участника совпадает с исходным ИНН, иначе false, то есть вычислялось с помощью калькулятора по формуле 4:

$$= IF(providerInn = providerInn\_add, 1{,}0), \quad (4)$$

где $providerInn$ – исходные значения ИНН участника, $providerInn\_add$ - добавленные значения.

Таким образом, из исходного набора данных (4607 записей) путем добавления к каждой закупке всех уникальных поставщиков стало 2998548 записей.

3.4 Исследование значимости признаков

Для определения значимости признаков используется коэффициент WoE-анализа – каждому наблюдению, содержащему набор признаков, ставится в соответствие бинарная выходная переменная (Событие или Не-событие). Затем весь диапазон изменения признака разбивается на несколько начальных классов, и для каждого вычисляется коэффициент WoE по формуле 5 [8]:

$$WoE_i = \ln \frac{F^-}{F^+}, \quad (5)$$

где i - индекс начального класса, F^-- относительная частота появления Не-событий в классе, F^+- относительная частота появления Событий в классе.

По WoE коэффициентам определяется информационный индекс (коэффициент IV) – величина, характеризующая значимость признака в модели бинарной классификации. Информационный индекс вычисляется по формуле 6:

$$IV = \sum_{i=1}^{k} \left\{ \left(\frac{N_i}{N} - \frac{P_i}{P} \right) * WoE_i \right\}, \quad (6)$$

где k – индекс начального класса, P_i – число событий, попавших в класс, P – общее число событий, N_i – число не-событий, попавших в интервал, N – общее число не-событий в исходном наборе данных.

Коэффициент IV всегда положителен, с его помощью определяется значимость признака [9]:

- отсутствует: IV < 0,02;
- низкая: 0,02 ⩽ IV < 0,1;
- средняя: 0,1 ⩽ IV < 0,3;
- высокая: IV > 0,3.

WoE-анализ используется в компоненте Конечные классы. Данный обработчик предназначен для [6]:

- преобразования дискретных и непрерывных входных полей, которые используются для построения моделей бинарной классификации, что, позволяет повысить точность и устойчивость модели прогнозирования к изменению входных данных;
- восстановления пропусков и упрощения описания исследуемых объектов;
- сокращение размерности данных за счет исключения признаков с низкой значимостью;
- борьбы с выбросами и экстремальными значениями.

Для формирования конечных классов критерием являются коэффициенты WoE и IV [9]:

- за счет максимизации значимости признака в бинарной классификационной модели;
- с помощью максимизации равномерности заполнения интервалов;
- с учетом первого и второго пункта.

В процессе работы обработчика Конечные классы производится преобразование входных столбцов в последовательность интервалов (конечные классы) с определенными метками. Для всех входных столбцов определеяется уровень значимости (отсутствует, очень низкая, низкая, средняя, высокая и очень высокая) и информационный индекс. Уровень значимости и информационный индекс помогают в отборе переменных для модели бинарной классификации. Что приводит к уменьшению количества значений переменной без ущерба для информативности данных [7].

Рассмотрим подробнее входные и выходные столбцы компонента Конечные классы. На входе: таблица данных и возможность добавления еще одного порта с диапазоном квантования [10].

На выходе:

1 порт - выходной набор данных (таблица данных). Структура данных:

- Поля исходного набора данных (значения не изменяются).
- <Метка столбца> Номер класса – идентификатор конечного класса, целое число (начиная с 0) – столбец создается всегда.
- <Метка столбца> Метка – метка конечного класса, полученная автоматическим путем (числовые границы, если это непрерывная переменная, или перечисление уникальных значений через «;», если переменная дискретная).
- <Метка столбца> Значимость.

2 порт - параметры классов (таблица данных). Структура данных:

- Группа – номер группы, к которой относится запись в таблице. Каждая группа записей ассоциирована с признаком (полем) исходного набора данных, являющимся входным для узла Конечные классы. Количество записей в группе соответствует числу конечных классов исходного столбца.
- Идентификатор – имя столбца, под которым он будет обрабатываться в наборе данных. Число столбцов равно числу входных полей узла Конечные классы.
- Метка столбца – мнемоническое обозначение входного столбца, под которым он будет виден пользователю в базе или хранилище данных. По умолчанию устанавливается название, под которым данный столбец виден в исходном наборе данных.
- Номер класса – порядковый номер (идентификатор), присвоенный классу при его формировании в узле Конечные классы.
- Уник.значение – для дискретных полей отображает их уникальные значения.
- Метка класса – метка класса, присвоенная ему при формировании в узле Конечные классы. Для числовых столбцов метка класса состоит из верхней и нижней границ класса (для нулевого класса указывается только нижняя граница с предлогом «от…», для класса с максимальным номером указывается верхняя граница с предлогом «до…»). Для категориальных полей, если каждый класс формируется для отдельной категории, то в качестве метки указывается эта категория. Если класс включает несколько категорий, то в метке перечисляются все категории, вошедшие в класс.
- Число событий – количество наблюдений в классе, для которых выходное значение – событие.
- Число не-событий – количество наблюдений в классе, для которых выходное значение – не-событие.
- Доля событий – отношение Числа событий к общему количеству Числа событий и Числа не-событий.

- Доля не-событий – отношение Числа не-событий к общему количеству Числа событий и Числа не-событий.
- Нижняя граница – для числовых признаков указывается нижняя граница интервала числом. Для категориальных признаков нижняя граница обозначается двумя категориями – верхней категорией предыдущего класса и нижней категорией текущего.
- Верхняя граница – для числовых признаков указывается верхняя граница интервала числом. Для категориальных признаков верхняя граница обозначается двумя категориями – нижней категорией следующего класса и верхней категорией текущего.
- Вес доказательства – коэффициент WoE для каждого класса.
- Информационный индекс – указываются значения информационных индексов IV, вычисленные по каждому входному столбцу. Сумма частных информационных индексов по каждому классу дает общий информационный индекс признака, по которому определяется его значимость.
- Доля класса – отношение количества наблюдений в классе к общему числу наблюдений.
- Верхняя граница диапазонов открыта.
- Предквантование – показывает применялось ли предквантование в процессе формирования конечных классов.

3 порт - значимости столбцов (таблица данных). Структура данных:

- Имя столбца – идентификатор столбца, под которым он будет обрабатываться в наборе данных. Число столбцов равно числу входных полей узла Конечные классы.
- Метка столбца – мнемоническое обозначение входного столбца, под которым он будет виден пользователю в базе или хранилище данных. По умолчанию устанавливается название, под которым данный столбец виден в исходном наборе данных.
- Число событий – количество событий, попавших в данный класс.

- Число не-событий – количество не-событий, попавших в данный класс.
- Доля событий – отношение Числа событий к общему количеству Числа событий и Числа не-событий.
- Доля не-событий – отношение Числа не-событий к общему количеству Числа событий и Числа не-событий.
- Всего – общее число наблюдений в классе.
- Информационный индекс – указываются значения информационных индексов IV, вычисленные по каждому входному столбцу.
- Значимость столбца – уровень значимости входного столбца, определенный на основе Информационного индекса. Может принимать значения:
    - отсутствует;
    - очень низкая;
    - низкая;
    - средняя;
    - высокая;
    - очень высокая.

Произведем оценку значимости сформированных признаков с помощью компонента Конечные классы: в назначении столбцов входными полями указываем все значения, кроме того, которого будем прогнозировать, то есть part - оно является выходным. Результат на рисунках 13 и 14.

Обратим внимание на значения коэффициента IV - везде, за исключением добавленного ИНН Победителя, данный коэффициент равен 0,00, что говорит о том, что значимые столбцы отсутствуют. Но, если посмотреть на соотношение событий к не-событиям (2925125 и 3423, соответственно), то можно сделать вывод о наличии дисбаланса классов. Для дальнейшей работы необходимо найти методы решения проблемы дисбаланса классов. Результаты WoE-анализа свидетельствуют о том, что соотношение

числа не-событий к числу событий для определенных ИНН участника меньше, то есть для них выше вероятность участия.

Метка столбца	Число событий	Число не-событий	Доля событий	Доля не-событий	Всего	Информационный индекс	Значимость столбца
Разница	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
Месяц закупки	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
Сезон	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
ИНН Победителя	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
Дата публикации	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
okved_new	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
customerinn	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
winner	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
date	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
count_uniq_prov_on_okved	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
count_part_prov	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
count_uniq_okv_on_prov	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
count_win	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
count_win_on_okved	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
min_price_on_okved	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
max_price_on_okved	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
lot_num	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
price_lot	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
COL2	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
min_date_prov	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
max_date_prov	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
purchaseNumber	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
okved	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
purchaseType	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
title	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
Разница_закупка_контракт	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
contactDateTime	2 995 125	3 423	1,00	0,00	2 998 548	0,00	Отсутствует
ИНН Победителя доб	2 995 125	3 423	1,00	0,00	2 998 548	1,27	Высокая

Рисунок 13 – Результат обработчика Конечные классы на исходных данных.

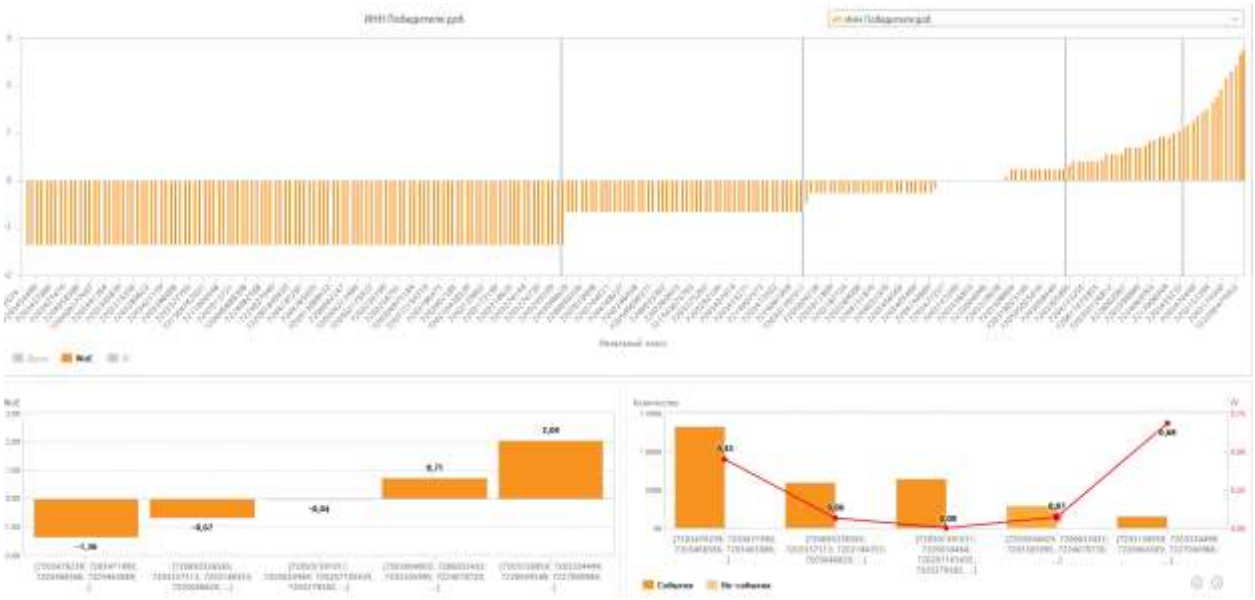

Рисунок 14 – Результат WoE-анализа на обучающей выборке.

Глава 4. Разработка модуля прогнозирования участников

4.1 Методы избавления от дисбаланса классов

Дисбаланс классов происходит, когда классы не представлены одинаково в задачах классификации и часто приводит к неточным и сниженным прогностическим характеристикам [10].

Для уменьшения дисбаланса классов традиционными методами считаются методы повторной выборки, которые делятся на две категории [11]: недостаточная выборка для мажоритарного класса и избыточная выборка для миноритарного класса.

К методам работы с недостаточной выборкой для мажоритарного класса относят [12]:

1. Random Under-sampling (RUS): случайным образом балансируются наблюдения между классами, путем удаления элементов мажоритарного класса.
2. Neighborhood cleaning rule (NCR): все наблюдения классифицируются по правилу трех ближайших соседей, затем удаляются следующие наблюдения мажоритарного класса:
    a. Которые получили верную метку.
    b. Которые являются соседями миноритарного класса и были неверно классифицированы.

К методам работы с избыточной выборкой для миноритарного класса можно отнести:

1. SMOTE: выбирается одно наблюдение из миноритарного класса, по отношению к нему определяется n-ближайших соседей из того же класса, далее из этих соседей случайным образом выбирается одно наблюдение и вычисляется эвклидово-расстояние между выбранными наблюдениями для каждого признака. Полученные расстояния умножаются на случайное число ($0 < r < 1$), затем найденные значения прибавляются к ранее выбранному n-му ближайшему соседу.

2. Random Over-sampling (ROS): генерирует случайным образом синтетические наблюдения миноритарного класса. Количество элементов миноритарного класса увеличивается до количества мажоритарного класса [13].

Далее методы дисбаланса классов будут рассмотрены подробнее на нашем примере. На рисунке 15 представлена часть схемы, где производилось тестирование методов избавления от дисбаланса классов. В приведенной на рисунке 15 схеме производится логическое разделение на четыре ветви, три из них соответствуют тестируемым методам Random over-sampling, Random under-sampling, RandomOverSampling, последняя – результаты на исходных данных. Каждая из ветвей, за исключением последней, содержит python-скрипт, с помощью которого производится работа над дисбалансом классов, следом, с помощью компонента Калькулятор, выполняется преобразование выходного поля в логический формат, далее – оценка значимых признаков и прогнозирование. Последняя ветвь состоит из оценки значимых признаков и прогнозирования на основе исходных данных, то есть с дисбалансом классов.

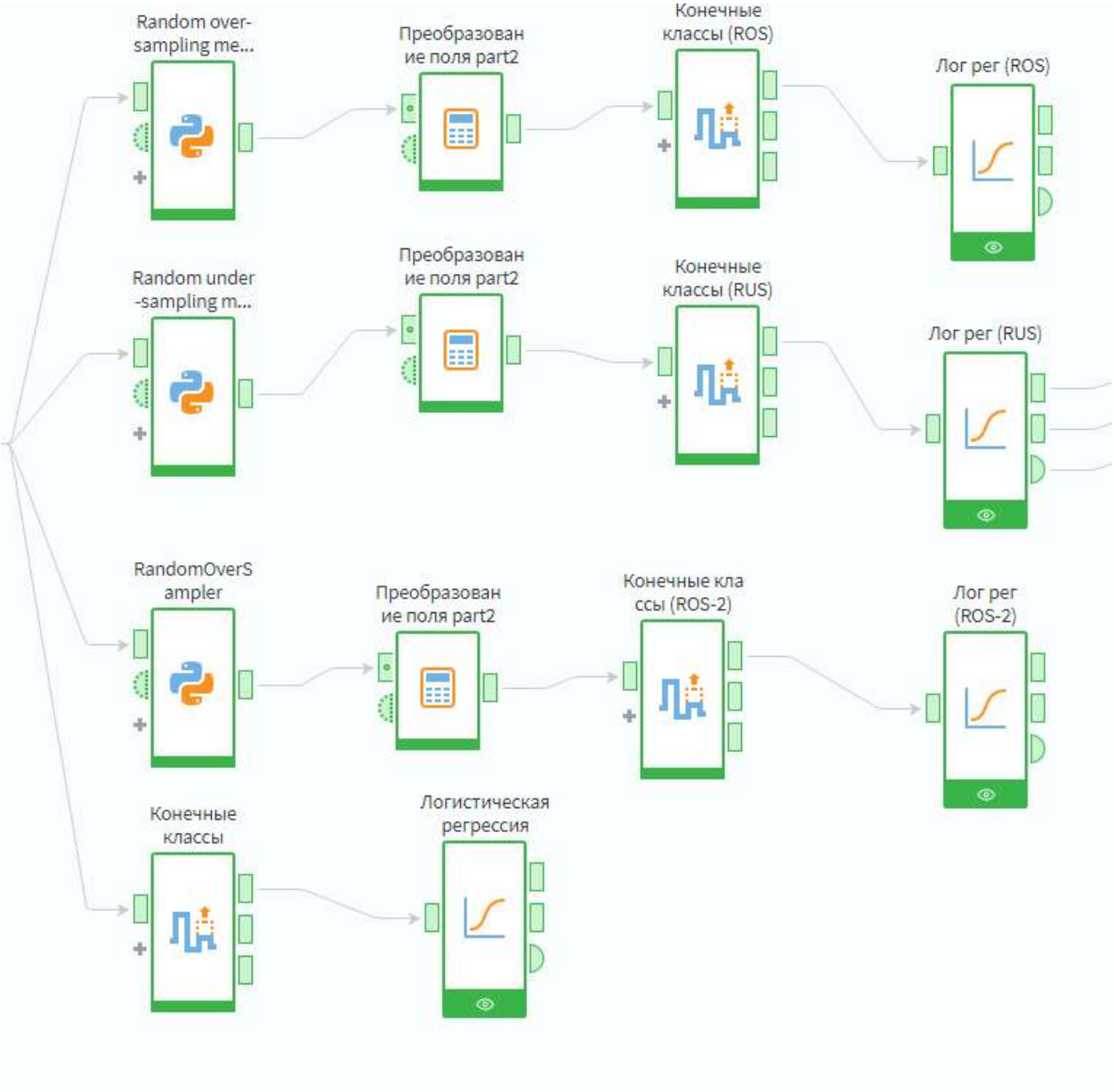

Рисунок 15 – Тестирование методов работы с дисбалансом классов.

4.1.1 Random over-sampling

Для использования данного метода был применен python-скрипт (приложение Б) [14]. Алгоритм работы данного метода заключается в дублировании случайных записей из класса меньшинства до того момента, пока их количество не станет равным количеству записей из мажоритарного класса. Исходный набор данных преобразовывается в dataframe, после чего он разбивается на две выборки, исходя из значений прогнозируемого поля part (0 или 1). Далее из выборки, содержащей элементы миноритарного класса (в текущем случае part = 1) случайным образом выбираются записи, в

количестве, равном разнице между количеством записей из класса меньшинств и количеством записей мажоритарного класса и конкатенируются к выборке, содержащей элементы мажоритарного класса (part = 0).

После чего с помощью компонента Конечные классы выполнялась оценка значимости признаков. Результаты представлены на рисунке 16.

Метка столбца	Число событий	Число не-событий	Доля событий	Доля не-событий	Всего	Информационный индекс
winner	1 648 500	4 341 750	0,28	0,72	5 990 250	10,42
time_between	1 648 500	4 341 750	0,28	0,72	5 990 250	0,06
month	1 648 500	4 341 750	0,28	0,72	5 990 250	0,01
season	1 648 500	4 341 750	0,28	0,72	5 990 250	0,01
publicationDateTime	1 648 500	4 341 750	0,28	0,72	5 990 250	0,04
okved_new	1 648 500	4 341 750	0,28	0,72	5 990 250	1,16
customerInn	1 648 500	4 341 750	0,28	0,72	5 990 250	0,09
date	1 648 500	4 341 750	0,28	0,72	5 990 250	0,04
count_uniq_prov_on_okved	1 648 500	4 341 750	0,28	0,72	5 990 250	0,02
count_part_prov	1 648 500	4 341 750	0,28	0,72	5 990 250	0,03
count_uniq_okv_on_prov	1 648 500	4 341 750	0,28	0,72	5 990 250	0,02
count_win	1 648 500	4 341 750	0,28	0,72	5 990 250	0,35
count_win_on_okved	1 648 500	4 341 750	0,28	0,72	5 990 250	0,00
min_price_on_okved	1 648 500	4 341 750	0,28	0,72	5 990 250	4,36
max_price_on_okved	1 648 500	4 341 750	0,28	0,72	5 990 250	4,33
lot_num	1 648 500	4 341 750	0,28	0,72	5 990 250	0,00
price_lot	1 648 500	4 341 750	0,28	0,72	5 990 250	9,60
COL2	1 648 500	4 341 750	0,28	0,72	5 990 250	1,16
min_date_prov	1 648 500	4 341 750	0,28	0,72	5 990 250	0,09
max_date_prov	1 648 500	4 341 750	0,28	0,72	5 990 250	0,05
purchaseNumber	1 648 500	4 341 750	0,28	0,72	5 990 250	5,50
okved	1 648 500	4 341 750	0,28	0,72	5 990 250	2,23
purchaseType	1 648 500	4 341 750	0,28	0,72	5 990 250	0,00
title	1 648 500	4 341 750	0,28	0,72	5 990 250	4,78
time_between_pur_contract	1 648 500	4 341 750	0,28	0,72	5 990 250	0,03
contacDateTime	1 648 500	4 341 750	0,28	0,72	5 990 250	0,05
providerInn_add	1 648 500	4 341 750	0,28	0,72	5 990 250	0,65
part	1 648 500	4 341 750	0,28	0,72	5 990 250	11,80

Рисунок 16 – Значимость признаков после дисбаланса классов с помощью метода Random over-sampling.

По рисунку 16 можно заметить, что доля событий по отношению к доле не-событий увеличилась и информационный индекс признаков вырос, а значит можно отобрать значимые признаки, которые будут использоваться для прогнозирования. Для этого, с помощью фильтра для полученных признаков ставим порог отсечения 0,1 (значимость столбца средняя и высокая). Таким образом, при прогнозировании будут учитываться следующие значения следующих столбцов (рисунок 17):

Имя столбца	Число событий	Число не-событий	Доля событий	Доля не-событий	Информацио...	Значимость ст...
winner	136 964	346 646	0,28	0,72	8,57	Высокая
okved_new	136 964	346 646	0,28	0,72	1,65	Высокая
count_win	136 964	346 646	0,28	0,72	0,75	Высокая
min_price_on_okved	136 964	346 646	0,28	0,72	5,77	Высокая
max_price_on_okved	136 964	346 646	0,28	0,72	5,50	Высокая
price_lot	136 964	346 646	0,28	0,72	8,32	Высокая
COL2	136 964	346 646	0,28	0,72	1,65	Высокая
purchaseNumber	136 964	346 646	0,28	0,72	7,42	Высокая
okved	136 964	346 646	0,28	0,72	3,55	Высокая
title	136 964	346 646	0,28	0,72	6,67	Высокая
part	136 964	346 646	0,28	0,72	10,20	Высокая

Рисунок 17 – Признаки прогнозирования, которые будут использоваться в прогнозировании.

Рассмотрим результаты WoE-анализа по некоторым признакам, которые будут использоваться в модели: общее количество побед поставщика и стоимость лота.

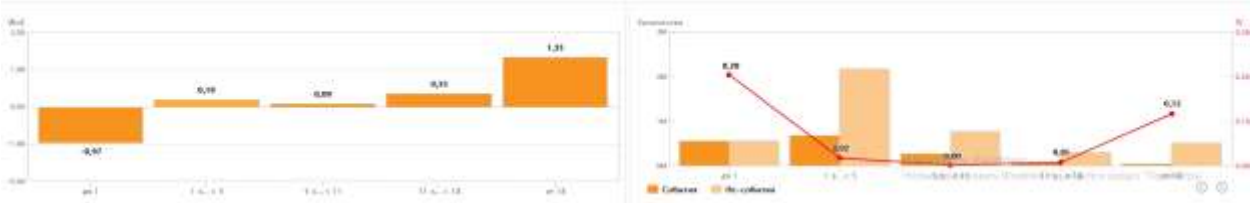

Рисунок 18 – Влияние общего количества побед поставщика на вероятность участия.

Согласно диаграммам, на рисунке 18, можно сделать выводы о влиянии количества побед поставщика на вероятность участия: если количество побед менее 1 – вероятность участия минимальна, но чем больше количество побед, тем вероятнее участие поставщика. Результаты, представленные на рисунке 19, свидетельствуют о том, что для некоторых значений стоимости лота вероятность участия близка к 100% (значения, выделенные синим блоком), для значений из зеленого блока – вероятность участия 90%.

Рисунок 19 – Влияние стоимости лота на вероятность участия.

4.1.2 Random under-sampling

В рамках данного метода работы с дисбалансом классов использовался python-скрипт, представленный в приложении В. Алгоритм работы данного метода заключается в случайном удалении записей из мажоритарного класса, пока их количество не станет равным количеству записей из миноритарного класса. После чего, с помощью компонента Конечные классы выполнялась оценка значимости признаков (результаты на рисунке 20).

Метка столбца	Число событий	Число не-событий	Доля событий	Доля не-событий	Информационный индекс	Значимость столбца
winner	1 891	4 955	0,28	0,72	6,21	Высокая
time_between	1 891	4 955	0,28	0,72	0,06	Низкая
month	1 891	4 955	0,28	0,72	0,01	Отсутствует
season	1 891	4 955	0,28	0,72	0,01	Отсутствует
providerinn	1 891	4 955	0,28	0,72	2,17	Высокая
publicationDateTime	1 891	4 955	0,28	0,72	0,06	Низкая
okved_new	1 891	4 955	0,28	0,72	0,38	Высокая
customerinn	1 891	4 955	0,28	0,72	0,08	Низкая
date	1 891	4 955	0,28	0,72	0,05	Низкая
count_uniq_prov_on_okved	1 891	4 955	0,28	0,72	0,03	Низкая
count_part_prov	1 891	4 955	0,28	0,72	0,04	Низкая
count_uniq_okv_on_prov	1 891	4 955	0,28	0,72	0,03	Низкая
count_win	1 891	4 955	0,28	0,72	0,34	Высокая
count_win_on_okved	1 891	4 955	0,28	0,72	0,00	Отсутствует
min_price_on_okved	1 891	4 955	0,28	0,72	2,21	Высокая
max_price_on_okved	1 891	4 955	0,28	0,72	2,14	Высокая
lot_num	1 891	4 955	0,28	0,72	0,00	Отсутствует
price_lot	1 891	4 955	0,28	0,72	7,01	Высокая
COL2	1 891	4 955	0,28	0,72	0,58	Высокая
min_date_prov	1 891	4 955	0,28	0,72	0,07	Низкая
max_date_prov	1 891	4 955	0,28	0,72	0,07	Низкая
purchaseNumber	1 891	4 955	0,28	0,72	4,27	Высокая
okved	1 891	4 955	0,28	0,72	1,53	Высокая
purchaseType	1 891	4 955	0,28	0,72	0,00	Отсутствует
title	1 891	4 955	0,28	0,72	3,21	Высокая
time_between_pur_contract	1 891	4 955	0,28	0,72	0,03	Низкая
contacDateTime	1 891	4 955	0,28	0,72	0,05	Низкая
providerinn_add	1 891	4 955	0,28	0,72	1,79	Высокая
part	1 891	4 955	0,28	0,72	7,13	Высокая

Рисунок 20 – Значимость признаков после дисбаланса классов с помощью метода Random under-sampling.

На рисунке 20 видно, что общее число событий сократилось за счет удаления записей мажоритарного класса, а также значительно увеличился информационный индекс, по сравнению с результатами анализа исходных данных (рисунок 13). Далее был произведен отбор значимых признаков с индексом информативности выше 0,1 (значимость столбца средняя и высокая).

4.1.3 Random OverSampler

Для использования random OverSampler был применен python-скрипт (приложение Г) [12]. Данный метод идентичен random over-sampling, но он относится к классу imblearn.over_sampling. Далее с помощью компонента Конечные классы выполнялась оценка значимости признаков и прогнозирование с использованием компонента логистическая регрессия. Результат оценки значимости классов идентичен результатам после применения random over-sampling (пункт 4.1.1).

4.2 Выбор модели прогнозирования

Модель прогнозирования – функциональное представление, которое описывает исследуемый процесс и является основой для получения его будущих значений.

Метод прогнозирования – совокупность последовательных действий, выполняющихся для получения модели прогнозирования.

Классификация методов прогнозирования представлена на блок-схеме (рисунок 21) [13]:

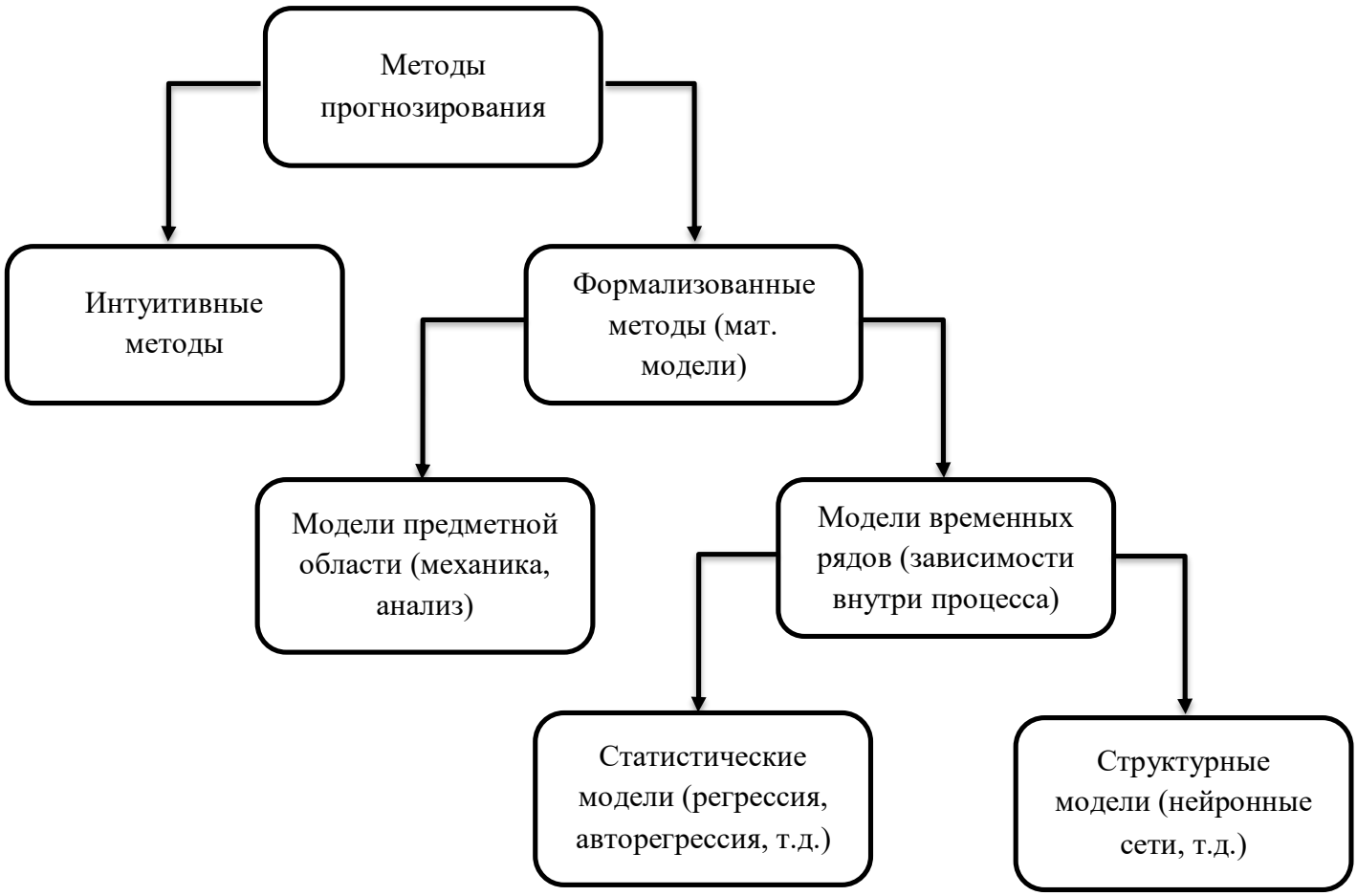

Рисунок 21 – Классификация методов и моделей прогнозирования.

К статическим моделям [15] относятся:

- регрессионные модели – линейная регрессия, нелинейная регрессия;

- модель экспоненциального сглаживания;
- авторегрессионные модели – ARIMAX, ARDLM и другие;
- модель по выборке максимального подобия.

Зависимость будущего значения от прошлого задается в виде некоторой структуры и правил перехода по ней в структурных моделях. К ним относятся, например, нейросетевые модели, модели на базе цепей Маркова, модели на базе классификационно-регрессионных деревьев.

Текущая модель не является моделью временных рядов, поскольку будет выполняться прогнозирование не объектов, а вероятности. Модель является моделью классификации с бинарной переменной класса (выходная переменная, может принимать только два значения). Для построения таких моделей широко применяются такие методы, как машины опорных векторов, логистическая регрессия и др. [17]. Линейный классификатор не подходит, поскольку есть необходимость получить вероятность принадлежности класса.

По данным различных исследований, наиболее распространенными моделями бинарного выбора являются логистическая регрессия и пробит-регрессия. Поэтому, в качестве инструмента исследования был выбран аппарат бинарной логистической регрессии.

Модель бинарной логистической регрессии (если более одной независимой переменной), имеет вид (формула 7):

$$P = \frac{1}{1-e^{-y}}, \quad (7)$$

где P – вероятность того, что произойдет интересующее событие, y – стандартное уравнение регрессии, e = 2,71.

В бинарной логистической регрессии зависимая переменная принимает только одно из двух значений (наступление или ненаступление события) [19].

4.3 Показатели качества прогноза

Качество прогноза оценивается [18] с помощью:

1. Коэффициенты для оценки качества моделей

a. Коэффициент детерминации – показывает процент объясненной моделью дисперсии [19]. Коэффициент детерминации наиболее популярен в регрессионном анализе, считается по обучающей части выборки, то есть показывает насколько хорошо были описаны данные, но абсолютно не гарантирует точность прогноза [20]. Рассчитывается по формуле 8:

$$R^2 = 1 - \frac{SSE}{TSS}, \quad (8)$$

где $SSE = \sum_{t=1}^{T} e_t^2$ – это сумма квадратов ошибок модели, $TSS = \sum_{t=1}^{T}(y_t - \bar{y})^2$- сумма квадратов отклонений фактических значений от средней величины.

b. Матрица ошибок – матрица, ij-я позиция, которой равна числу объектов i-го класса, которым алгоритм присвоил метку j-го класса (размер 2×2). Классы делятся на положительный (метка – 1) и отрицательный (0 или – 1). Объекты, с меткой 1, называются положительными (Positive), те из них, которые на самом деле принадлежат к этому классу – истинно положительными (True Positive - TP), остальные – ложно положительными (False Positive - FP). Аналогично есть для отрицательного (Negative) класса.

c. Точность (Accuracy) - доля объектов, для которых алгоритм выдал правильные ответы. Точность некорректно использовать в случае дисбаланса классов (когда представителей одного из класса существенно больше, чем другого). Рассчитывается по формуле 9:

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m} I[a_i = y_i], \quad (9)$$

где y_i – метка i-го объекта, a_i – ответ на этом объекте нашего алгоритма, m – число объектов в выборке.

Также, accuracy можно вычислить по формуле 10:

$$Accuracy = \frac{TN+TP}{TN+FN+TP+FP}, \quad (10)$$

где TP - истинно положительные, TR – истинно отрицательные, FP – ложно положительные, FN – ложно отрицательные.

d. Полнота – какая доля объектов, реально относящихся к положительному классу, была предсказана верно. Определяется по формуле 11:

$$TPR = \frac{TP}{TP+FN}, \quad (11)$$

где TP - истинно положительные, FN – ложно отрицательные.

e. Точность (Precision) – какая доля объектов, распознанных как объекты положительного класса, была предсказана верно. Вычисляется по формуле 12:

$$PPV = \frac{TP}{TP+FP}, \quad (12)$$

где TP - истинно положительные, FP – ложно положительные.

2. ROC-анализ.

ROC-кривая - кривая, которая показывает зависимость количества верно классифицированных положительных примеров (истинно положительное множество) от количества неверно классифицированных отрицательных примеров (ложно отрицательное множество) [21]. Является самым наглядным методом оценки качества модели бинарной классификации.

Показатель AUC – площадь, ограниченная ROC-кривой и осью доли ложных положительных классификаций. Чем выше AUC, тем качественнее классификатор.

Если AUC равен 0,5 – это говорит о непригодности выбранного метода классификации (случайное гадание). Экспертная шкала для оценки AUC (качества прогноза) представлена в таблице 3.

Таблица 3 – Шкала для оценки AUC

Интервал AUC	Качество модели
0,9-1,0	Отличное
0,8-0,9	Очень хорошее
0,7-0,8	Хорошее
0,6-0,7	Среднее
0,5-0,6	Неудовлетворительное

4.4 Построение модели прогнозирования

Логистическая регрессия – это один из видов множественной регрессии, которая предназначена для анализа связи между несколькими независимыми переменными и зависимой переменной. Бинарная логистическая регрессия применяется в случае, когда зависимая переменная может принимать только два значения. Логистической регрессии позволяет оценивать вероятность того, что событие наступит.

Важная особенность логистической регресии:

- входные данные никогда не должны содержать пропусков;
- выходные данные не должны содержать пропусков во время обучения.

Обязательным входным полем компонента Логистическая регрессия является – входной источник данных.

Выход:

- выход регрессии. Таблица, состоящая из полей:
  - Событие|Прогноз.
  - Вероятность события|Прогноз.
  - Событие|Факт.
  - Поле выходных данных|Прогноз.
  - Все поля исходного набора данных.
- коэффициенты регрессионной модели:
  - имена входных полей;
  - метки входных полей;
  - уникальные значения;
  - коэффициенты;
  - стандартное отклонение;
  - критерий Вальда - оценка значимости коэффициента при независимой переменной модели;
  - отношение шансов – величина, которая определяет отношение вероятности события к не-событию.

- сводка.

Настройка параметров логистической регрессии включает в себя:

- отбор факторов и защита от переобучения:
    - принудительное включение (Enter) – включение в регрессионную модель всех входных признаков;
    - пошаговое включение (Forward) – метод, который начинает с отсутствия признаков и постепенно ищет самые "лучшие", которые будут добавлены в подмножество;
    - пошаговое включение/исключение (Stepwise) – на каждом шаге после включения новой переменной в модель осуществляется проверка на значимость остальных переменных, которые уже были введены в нее ранее (модификация метода Forward);
    - пошаговое исключение (Backward) – начинает со всех доступных признаков и исключает самые "плохие";
    - LASSO – метод для борьбы с переизбыточностью данных;
    - Ridge – метод понижения размерности, который применяется для борьбы с избыточными данными (независимые переменные коррелируют друг с другом, что приводит к неустойчивости оценок коэффициентов регрессии);
    - Elastic-Net — модель регрессии с двумя регуляризаторами L1, L2, которые помогают улучшить обобщение и ошибки теста, поскольку не допускают переобучения модели из-за шума в данных:
        - L1 — отбор наиболее важных факторов, сильнее всего влияющих на результат;
        - L2 — предотвращает переобучения модели с помощью запрета на непропорционально большие весовые коэффициенты.

- Приоритет точность/скорость (целочисленный тип в диапазоне от 0 до 4 включительно):
    - максимальная / повышенная точность;
    - средняя / повышенная / максимальная скорость.
- Приоритет точные/недостоверные данные (целочисленный тип в диапазоне от 0 до 4 включительно): точные данные, повышенная / средняя / пониженная точность или недостоверные данные.
- Приоритет меньше/больше факторов (целочисленный тип в диапазоне от 0 до 4 включительно):
    - минимум / максимум факторов;
    - меньше / больше факторов;
    - среднее число факторов.

Поскольку конечные классы на исходных данных (до работы с дисбалансом классов) не показали информационный индекс выше 0 (рисунок 11), для теста в логистической регрессии входными признаками были использованы значения: okved_new, okved, count_win, price_lot, min_price_lot_on_okved, max_price_lot_on_okved, а выходным - part. В настройках логистической регрессии задано разбиение на множества: 70% обучающая выборка, 30% тестовая выборка, с последовательным методом разбиения (поскольку будут прогнозироваться участники будущих закупок, на основе текущих данных). Встроенные методы валидации логистической регрессии нет необходимости использовать, потому как ранее была проведена работа с дисбалансом классов. Отбор факторов и защита от переобучения выбираются автоматически, порог отсечения – 0,5.

Результаты на рисунке 22, на рисунке 23 представлена ROC-кривая. AUC ROC на обучающем множестве 0,5664, на тестовом - 0,0830.

Имя	Метка	Значение
12 TotalSamples	Всего примеров	2 998 548
12 TotalSelectedSamples	Всего отобранных примеров	2 998 548
12 TrainSamples	Примеров в обучающем множестве	2 098 984
9.0 TrainRMSError	Среднеквадратическая ошибка на обучающем множестве	0,04
9.0 TrainClsErrorPercentage	Процент ошибок классификации на обучающем множестве	0,14
9.0 TrainAvgCE	Средняя перекрестная энтропия на обучающем множестве	0,02
9.0 TrainThreshold	Порог отсечения при обучении модели	0,50
9.0 MinusTwoLogL	-2 Логарифма функции правдоподобия	44 535,54
9.0 R2	Коэффициент детерминации	0,00
9.0 AdjustedR2	Скорректированный коэффициент детерминации	0,00
9.0 Chi2	Хи-квадрат	147,39
12 ModelDF	Число степеней свободы модели	4 667
9.0 AIC	Информационный критерий Акаике	0,03
9.0 AICc	Информационный критерий Акаике скорректированный	0,03
9.0 BIC	Информационный критерий Байеса	0,05
9.0 HQC	Информационный критерий Ханнана-Куинна	0,03
9.0 ModelPValue	P-значение модели	1,00
12 TestSamples	Примеров в тестовом множестве	899 564
9.0 TestRMSError	Среднеквадратическая ошибка на тестовом множестве	0,02
9.0 TestClsErrorPercentage	Процент ошибок классификации на тестовом множестве	0,05
9.0 TestAvgCE	Средняя перекрестная энтропия на тестовом множестве	0,01

Рисунок 22 – Результат логистической регрессии до работы с дисбалансом классов.

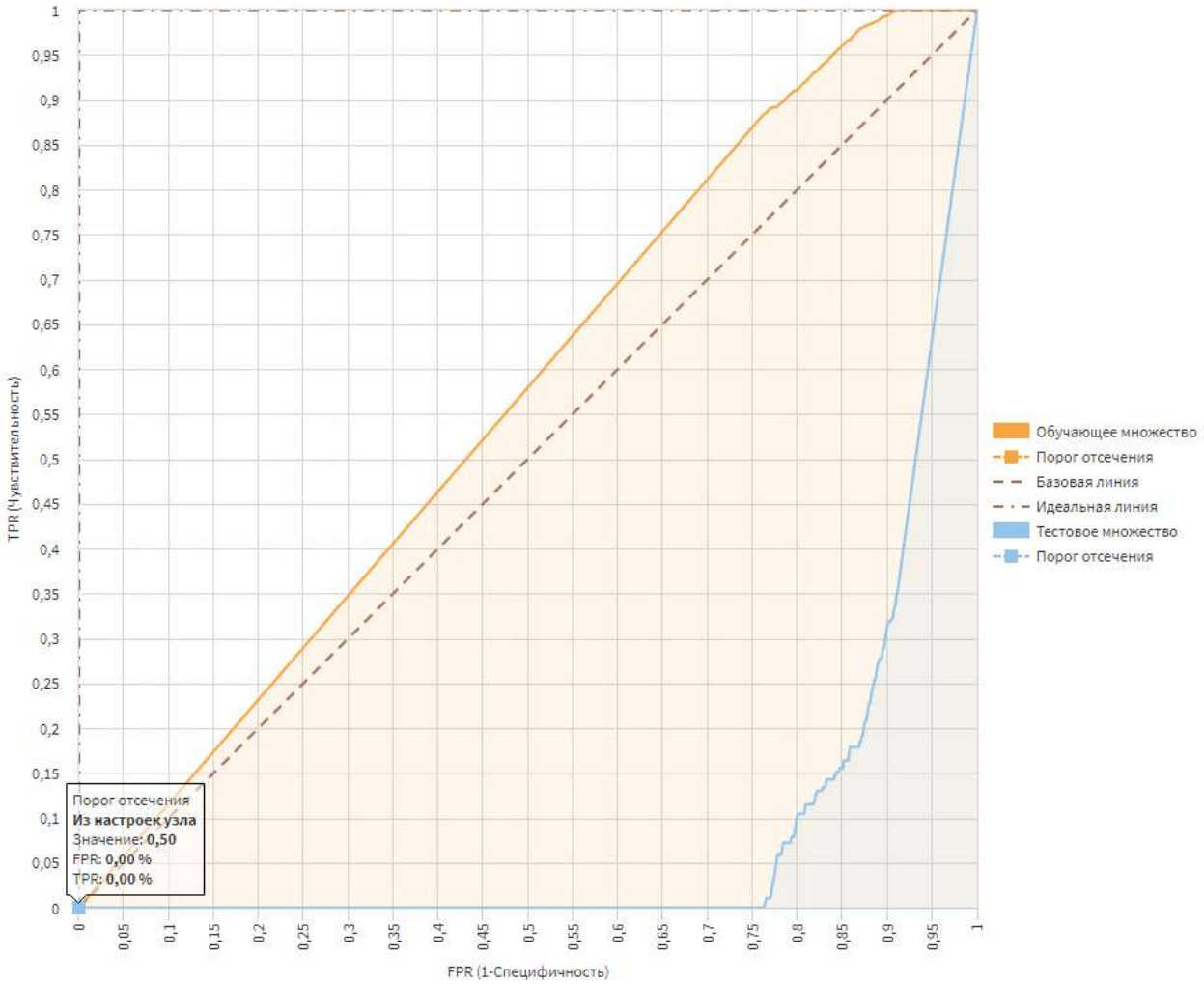

Рисунок 23 – ROC-кривая логистической регрессии до работы с дисбалансом классов.

4.4.1 Random over-sampling

Результаты работы компонента Конечные классы представлены на рисунке 15, признаки, которые необходимо использовать на входе изображены на рисунке 16.

Результаты прогнозирования представлены на рисунке 24.

Имя	Метка	Значение
12 TotalSamples	Всего примеров	5 990 250
12 TotalSelectedSamples	Всего отобранных примеров	5 990 250
12 TrainSamples	Примеров в обучающем множестве	4 193 175
9.0 TrainRMSError	Среднеквадратическая ошибка на обучающем множестве	0,40
9.0 TrainClsErrorPercentage	Процент ошибок классификации на обучающем множестве	27,50
9.0 TrainAvgCE	Средняя перекрестная энтропия на обучающем множестве	0,48
9.0 TrainThreshold	Порог отсечения при обучении модели	0,50
9.0 MinusTwoLogL	-2 Логарифма функции правдоподобия	2 767 183,80
9.0 R2	Коэффициент детерминации	0,51
9.0 AdjustedR2	Скорректированный коэффициент детерминации	0,51
9.0 Chi2	Хи-квадрат	2 852 773,05
12 ModelDF	Число степеней свободы модели	5 306
9.0 AIC	Информационный критерий Акаике	0,66
9.0 AICc	Информационный критерий Акаике скорректированный	0,66
9.0 BIC	Информационный критерий Байеса	0,68
9.0 HQC	Информационный критерий Ханнана-Куинна	0,67
9.0 ModelPValue	P-значение модели	0,00
12 TestSamples	Примеров в тестовом множестве	1 797 075
9.0 TestRMSError	Среднеквадратическая ошибка на тестовом множестве	0,40
9.0 TestClsErrorPercentage	Процент ошибок классификации на тестовом множестве	27,52
9.0 TestAvgCE	Средняя перекрестная энтропия на тестовом множестве	0,63

Рисунок 24 – Значения логистической регрессии после использования метода random over-sampling

AUC-ROC на обучающем множестве составила: 0,8124, на тестовом: 0,8124. ROC-кривая на рисунке 25, матрица ошибок - рисунок 26.

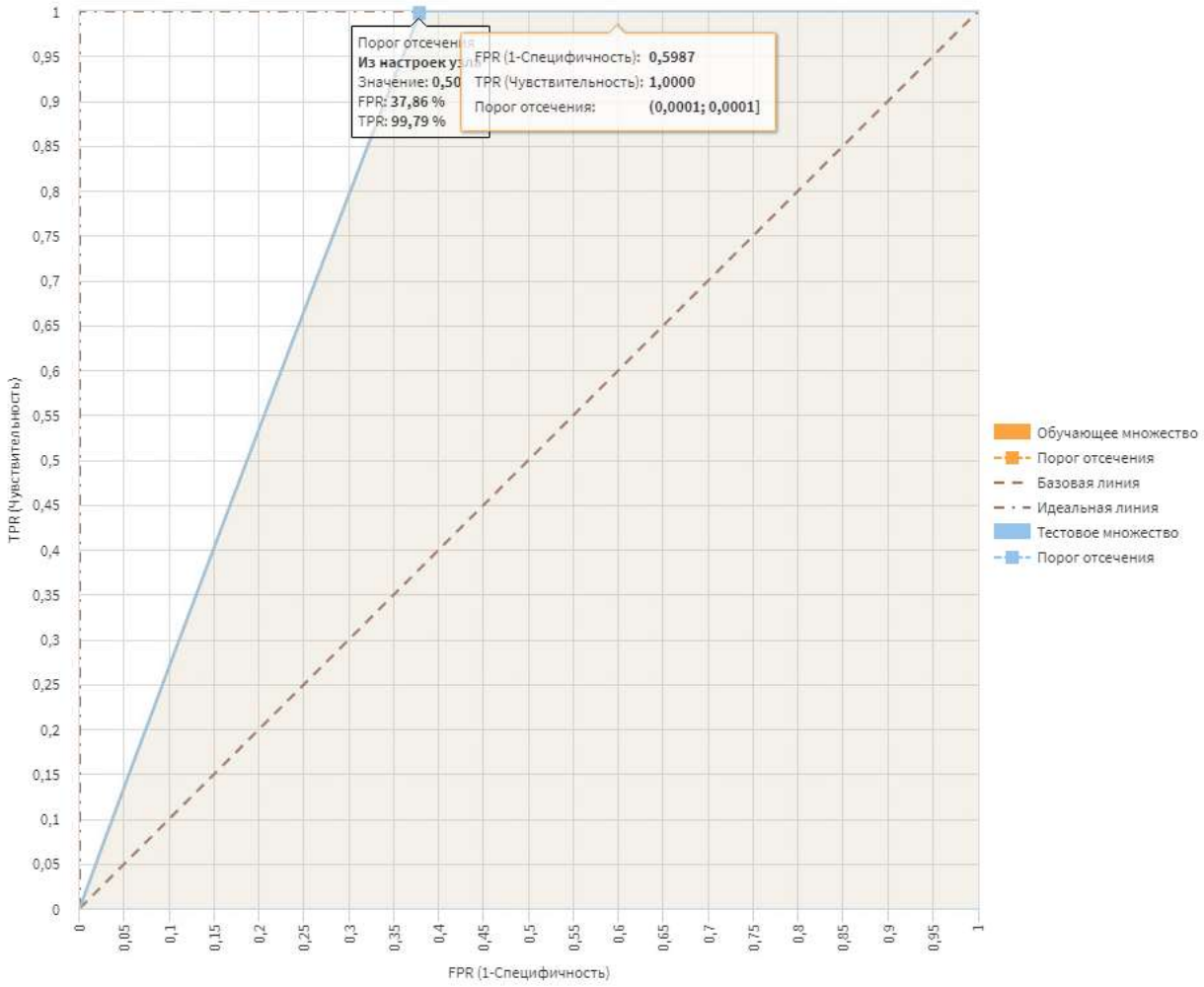

Рисунок 25 – ROC-кривая логистической регрессии после использования метода random over-sampling.

Классифицировано	Фактически		Итого
	Событие	Не-событие	
Обучающее	27,52%	72,48%	
Событие	27,46%	27,44%	54,90%
Не-событие	0,06%	45,04%	45,10%
Тестовое	27,52%	72,48%	
Событие	27,46%	27,46%	54,92%
Не-событие	0,06%	45,02%	45,08%

Рисунок 26 – Матрица ошибок после использования метода random over-sampling.

4.4.2 Random under-sampling

Значения компонента Конечные классы на рисунке 15. Результат значимости признаков после отсечения совпал с результатами значимости признаков после дисбаланса классов с помощью метода Random under-sampling, поэтому в логистической регрессии использовались такие же входные признаки. Показатели логистической регрессии представлены на рисунке 27, матрица ошибок - рисунок 28, ROC-кривая на рисунке 29. AUC-ROC на обучающем множестве составила: 0,8479, на тестовом: 0,8508.

Имя	Метка	Значение
12 TotalSamples	Всего примеров	6 846
12 TotalSelectedSamples	Всего отобранных примеров	6 846
12 TrainSamples	Примеров в обучающем множестве	4 792
9.0 TrainRMSError	Среднеквадратическая ошибка на обучающем множестве	0,45
9.0 TrainClsErrorPercentage	Процент ошибок классификации на обучающем множестве	23,81
9.0 TrainAvgCE	Средняя перекрестная энтропия на обучающем множестве	0,24
9.0 TrainThreshold	Порог отсечения при обучении модели	0,50
9.0 MinusTwoLogL	-2 Логарифма функции правдоподобия	1 597,42
9.0 R2	Коэффициент детерминации	0,75
9.0 AdjustedR2	Скорректированный коэффициент детерминации	0,75
9.0 Chi2	Хи-квадрат	4 831,22
12 ModelDF	Число степеней свободы модели	5 306
9.0 AIC	Информационный критерий Акаике	2,55
9.0 AICc	Информационный критерий Акаике скорректированный	0,00
9.0 BIC	Информационный критерий Байеса	9,72
9.0 HQC	Информационный критерий Ханнана-Куинна	5,07
9.0 ModelPValue	P-значение модели	1,00
12 TestSamples	Примеров в тестовом множестве	2 054
9.0 TestRMSError	Среднеквадратическая ошибка на тестовом множестве	0,45
9.0 TestClsErrorPercentage	Процент ошибок классификации на тестовом множестве	23,47
9.0 TestAvgCE	Средняя перекрестная энтропия на тестовом множестве	1,73

Рисунок 27 – Значения логистической регрессии после использования метода random under-sampling.

Классифицировано	Фактически		Итого
	Событие	Не-событие	
Обучающее	1 332	3 460	
Событие	1 235	1 044	2 279
Не-событие	97	2 416	2 513
Тестовое	559	1 495	
Событие	519	442	961
Не-событие	40	1 053	1 093

Рисунок 28 – Матрица ошибок логистической регрессии после использования метода random under-sampling.

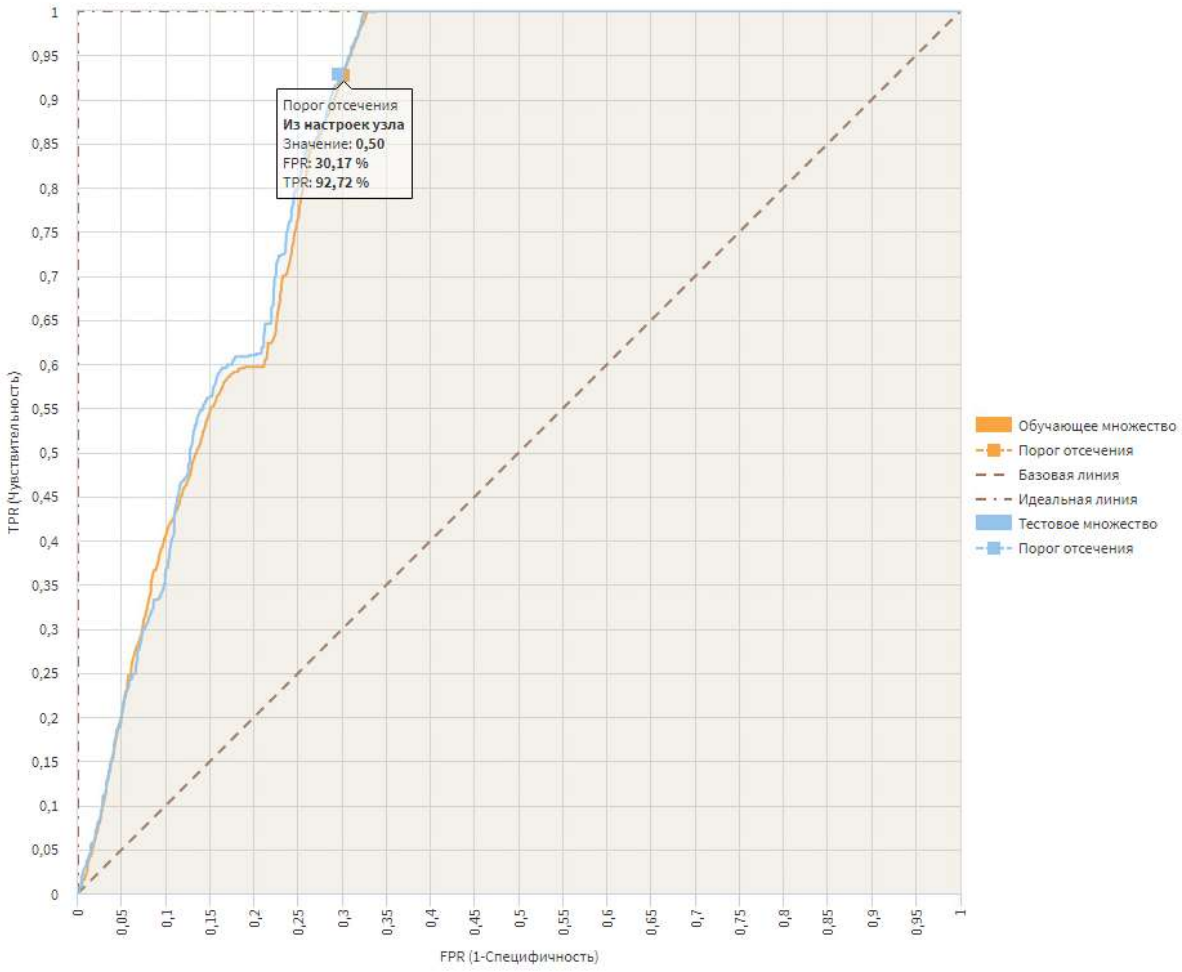

Рисунок 29 – ROC-кривая логистической регрессии после использования метода random under-sampling.

4.4.3 Random OverSampler

С помощью компонента Конечные классы выполнялась оценка значимости признаков и прогнозирование с использованием компонента логистическая регрессия. Результат оценки значимости классов идентичен результатам после применения random over-sampling. Показатели логистической регрессии представлены на рисунке 30, матрица ошибок - рисунок 31, ROC-кривая на рисунке 32. AUC-ROC на обучающем множестве составила: 0,8123, на тестовом: 0,8120.

Имя	Метка	Значение
12 TotalSamples	Всего примеров	5 990 250
12 TotalSelectedSamples	Всего отобранных примеров	5 990 250
12 TrainSamples	Примеров в обучающем множестве	4 193 175
9.0 TrainRMSError	Среднеквадратическая ошибка на обучающем множестве	0,40
9.0 TrainClsErrorPercentage	Процент ошибок классификации на обучающем множестве	27,54
9.0 TrainAvgCE	Средняя перекрестная энтропия на обучающем множестве	0,48
9.0 TrainThreshold	Порог отсечения при обучении модели	0,50
9.0 MinusTwoLogL	-2 Логарифма функции правдоподобия	2 768 543,77
9.0 R2	Коэффициент детерминации	0,51
9.0 AdjustedR2	Скорректированный коэффициент детерминации	0,51
9.0 Chi2	Хи-квадрат	2 851 413,08
12 ModelDF	Число степеней свободы модели	5 306
9.0 AIC	Информационный критерий Акаике	0,66
9.0 AICc	Информационный критерий Акаике скорректированный	0,66
9.0 BIC	Информационный критерий Байеса	0,68
9.0 HQC	Информационный критерий Ханнана-Куинна	0,67
9.0 ModelPValue	P-значение модели	0,00
12 TestSamples	Примеров в тестовом множестве	1 797 075
9.0 TestRMSError	Среднеквадратическая ошибка на тестовом множестве	0,40
9.0 TestClsErrorPercentage	Процент ошибок классификации на тестовом множестве	27,52
9.0 TestAvgCE	Средняя перекрестная энтропия на тестовом множестве	0,63

Рисунок 30 – Показатели логистической регрессии после использования метода Random OverSampling.

Классифицировано	Фактически		Итого
	Событие	Не-событие	
Обучающее	27,51%	72,49%	
Событие	27,45%	27,48%	54,93%
Не-событие	0,06%	45,01%	45,07%
Тестовое	27,55%	72,45%	
Событие	27,49%	27,46%	54,95%
Не-событие	0,06%	44,99%	45,05%

Рисунок 31 – Матрица ошибок логистической регрессии после использования метода Random OverSampling.

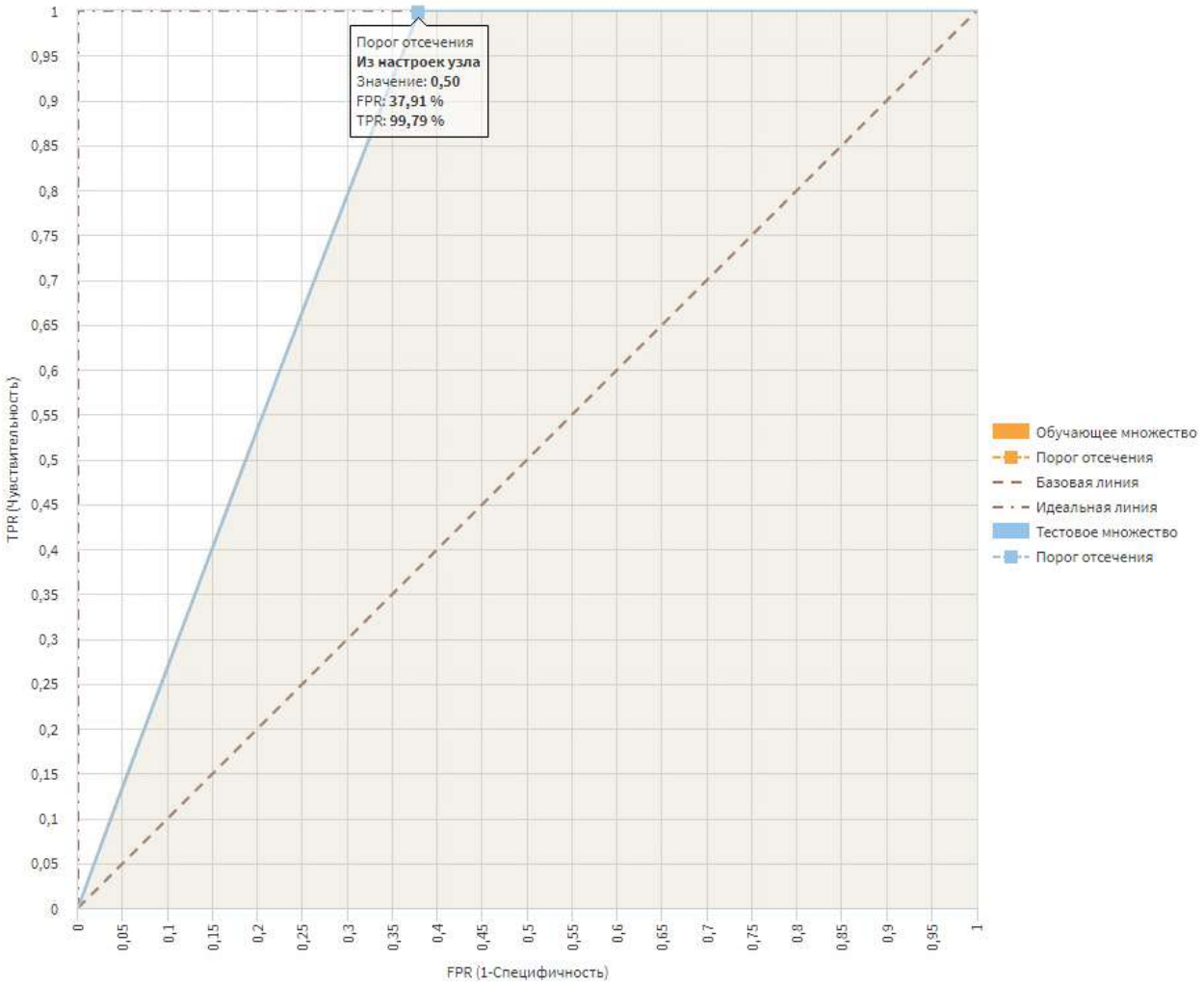

Рисунок 32 – ROC-кривая логистической регрессии после использования метода Random OverSampler.

4.4.4 Сравнительный анализ результатов прогнозирования

Таблица 4 – Сравнительный анализ результатов прогнозирования

Обработка дисбаланса классов	Общий объем данных	Доля не-событий	Ошибки на обучающем множестве, %	Ошибки на тестовом множестве, %	Коэффициент детерминации	AUC ROC
Исходные данные	2998548	0	0,14	0,05	0	Train - 0,5664, test - 0,0830
random over-sampling	5990250	0,72	27,50	27,52	0,51	Train - 0,8124, test - 0,8124
random under-sampling	6846	0,72	23,81	23,47	0,75	Train - 0,8479, test - 0,8505
Random OverSampler	5990250	0,72	27,54	27,52	0,51	Train - 0,8123, test - 0,8120

Из таблицы 4 видно, что результаты после методов random over-sampling и Random OverSampler схожи, поэтому далее будем рассматривать один из них (Random OverSampling) и random under-sampling. По всем показателям результаты прогнозирования после использования метода random under-sampling принимают лучшие значения: ROC-кривая (рисунок 27), показатель AUC ROC и количество ошибок. Согласно шкале для оценки AUC (таблица 3), значения AUC 0,85 (train - 0,8479, test - 0,8505) свидетельствуют об очень хорошем качестве прогноза.

4.5 Интеграция модуля прогнозирования и системы Метатендер

Веб-сервис – программная система со стандартизированными интерфейсами (которая идентифицируется уникальным веб-адресом), а также HTML-документ сайта, отображаемый браузером пользователя. Веб-службы

могут взаимодействовать как друг с другом, так и со сторонними приложениями с помощью SOAP, XML-RPC и REST.

В Loginom есть инструменты интеграции с внешними веб-сервисами и публикации своих собственных. Благодаря этому решения, созданные на базе Loginom, могут быть легко встроены в IT-ландшафт любой сложности. Интеграция и работа с веб-сервисами требует наличия следующих компонентов платформы Loginom:

- Server - основной компонент платформы. Выполняет задачи загрузки, расчетов, построения моделей, визуализации, управления правами и прочее.
- Integrator - компонент, отвечающий за публикацию собственных веб-сервисов на основе созданных в Loginom пакетов. Работает в связке с IIS, создавая в нем отдельное приложение.
- Adapter - необязательный компонент. Требуется для взаимодействия с нестандартными веб-сервисами, например, с бюро кредитных историй.

Есть возможность создания собственных SOAP и REST сервисов. При этом в публикуемом в качестве веб-сервиса узле пакета реализуется логика обработки данных запроса к веб-сервису. Чаще всего таким узлом является подмодель, поскольку в ней возможно реализовать произвольную логику обработки. Входные порты подмодели задают структуру запроса к сервису, а выходные – структуру ответа.

Взаимодействие с веб-сервисом осуществляется при помощи его методов. Каждый метод веб-сервиса имеет определенный идентификатор и выполняет сопоставленное этому идентификатору действие. Для каждого из публикуемого узлов создается соответствующий ему метод SOAP-сервиса и операция REST-сервиса. В случае с REST-сервисом для каждого из публикуемых пакетов создается конечная точка REST-сервиса.

Для работы с веб-сервисами необходима серверная редакциях Loginom при наличии установленного компонента Loginom Integrator. По техническим

причинам в ТюмГУ не развернута серверная редакция Loginom с компонентом Integrator.

Для серверной редакции установлены следующие требования: Windows Server 2012, 4-ядерный процессор, 4 Гб оперативной памяти, 500 Гб жесткого диска и ПО – .NET 4.5 и IIS 8.0 и выше. В дальнейшем планируется установить серверную редакцию Loginom на выделенной виртуальной машине с необходимыми требованиями и запустить веб-сервис для интеграции модуля прогнозирования и системы Метатендер.

ЗАКЛЮЧЕНИЕ

Разработан модуль прогнозирования участников государственных закупок по 44 федеральному закону, с помощью которого можно определять участников будущих закупок. Инструментарий реализован на базе аналитической платформы Loginom и включает в себя процедуры первичной загрузки и очистки данных (выявление и обработка пропусков, ошибок и противоречий), вычисление признаков прогнозирования, методы избавления от дисбаланса классов. Разработаны инструменты для определения следующих показателей прогнозирования:

1. Характеристики закупки:
    1.1. Количество уникальных участников по всем классификаторам закупок.
    1.2. Месяц публикации закупки.
    1.3. Сезон публикации закупки.
2. Характеристики участников:
    2.1. Количество уникальных классификаторов закупки, в которых принимал участие поставщик.
    2.2. Общее количество побед.
    2.3. Минимальная и максимальная дата участия у поставщика.
    2.4. Разница в месяцах между первой и последней датой публикации закупки, в которых принимал участие поставщик.
    2.5. Минимальная и максимальная стоимость лота у поставщика по каждому классификатору закупки.
    2.6. Количество участий и побед поставщика по каждому классификатору закупки.

После накопления данных о закупках в объеме, разработанный программный продукт предполагается дополнить моделью оценки вероятности победы участников в государственных закупках.

По итогам выполненной работы опубликована статья [22], а также готовится к публикации вторая статья.

СПИСОК ЛИТЕРАТУРЫ

1. Система Метатендер [Электронный ресурс] // metatender.ru: [Сайт]. URL: https://metatender.ru/ (дата обращения 03.03.2021).
2. Единая информационная система в сфере закупок: официальный сайт [Электронный ресурс] // zakupki.gov.ru: [Сайт]. URL: https://zakupki.gov.ru/epz/main/public/home.html (дата обращения 12.03.2021).
3. Аналитическая платформа Loginom: официальный сайт [Электронный ресурс] // help.loginom.ru: [Сайт]. URL: https://help.loginom.ru/userguide/index.html (дата обращения 15.03.2021).
4. Casari, A. Feature Engineering for Machine Learning. / A. Casari, A. Zheng. – New York: O'Reilly Media, 2018. - 218 p.
5. Конференция ММРО-15. Конкурс по анализу данных. Задача предсказания отклика клиентов ОТП банка [Электронный ресурс] // machinelearning.ru: [Сайт]. URL: http://www.machinelearning.ru/wiki/images/d/d5/Mmro-15_contest.pdf (дата обращения 21.03.2021).
6. Bari, A. Predictive Analytics. / A. Bari, M. Chaouchi, T. Yung. – New York: For Dummies, 2014. - 360 c.
7. Practical tips for class imbalance in binary classification [Электронный ресурс] // towardsdatascience.com: [Сайт]. URL: https://towardsdatascience.com/practical-tips-for-class-imbalance-in-binary-classification-6ee29bcdb8a7 (дата обращения 10.04.2021).
8. Кравцова, В. П. Основные аспекты моделирования вероятности отклика. / В. П. Кравцова // Научно-методический электронный журнал “Концепт”. - 2016. - №59 - С. 1-5.
9. Лагерев, Д. Г. Особенности построения скоринговой модели на основе аналитической платформы Deductor./ Д. Г. Лагерев, И. В. Бондарева // Научно-технический вестник Брянского государственного университета. - 2017. - №1 - С. 81-85.

10. Паклин, Н. Б. Оптимальное квантование для повышения качества бинарных классификаторов. / Н. Б. Паклин, В. В. Афанасьев // «Искусственный интеллект». - 2013. - №4 - С. 392-399.

11. Рюмина, Е. В. Сравнительный анализ методов устранения дисбаланса классов эмоций в видеоданных выражений лиц. / Е. В. Рюмина, А. А. Карпов // Научно-технический вестник информационных технологий, механики и оптики. - 2020. - №5 - С. 683-702.

12. Бардамова, М. Б. Методы предобработки несбалансированных классов. / М. Б. Бардамова // Сборник избранных статей научной сессии ТУСУР. - 2018. - №1 - С. 112-115.

13. Севастьянов, Л. А. О методах повышения точности многоклассовой классификации на несбалансированных данных. / Л. А. Севастьянов, Е. Ю. Щетинин // Информ. и её примен. - 2020. - №1 - С. 63–70.

14. Dr. Dataman. Using Over-Sampling Techniques for Extremely Imbalanced Data [Электронный ресурс] // medium.com: [Сайт]. URL: https://medium.com/dataman-in-ai/sampling-techniques-for-extremely-imbalanced-data-part-ii-over-sampling-d61b43bc4879 (дата обращения 10.05.2021).

15. Вьюгин, В. В. Математические основы машинного обучения и прогнозирования. / В.В. Вьюгин. - Москва: МЦНМО, 2013. - 304 с.

16. Хасти, Т. Основы статистического обучения: интеллектуальный анализ данных, логический вывод и прогнозирование. / Т. Хасти, Р. Тибришани, Д. Фридман – Москва: Вильямс, 2020. - 768 с.

17. Матраева, Л. В. Использование логистической регрессии при выявлении приоритетов региональной инвестиционной политики в отношении иностранных инвесторов в регионы РФ. / Л. В. Матраева // Статистика и математические методы в экономике. - 2013. - №6 - С. 170–174.

18. Егорова, С. В. Оценка точности экономических прогнозов: вопросы методики. / С. В. Егорова // Экономика и управление. - 2014. - №9 - С. 55–58.

19. Гржибовский, А. М. Корреляционный анализ. / А. М. Гржибовский // Экология человека. - 2008. - №9 - С. 50–60.

20. Кинякин, В. Н. Некоторые предостережения по проверке качества модели регрессии с помощью коэффициента детерминации. / В. Н. Кинякин, Ю. С. Милевская // Вестник Московского Университета МВД России. - 2014. - №8 - С. 200–204.

21. Кильдишев, Г.С. Анализ временных рядов и прогнозирование. / Г. С. Кильдишев – Москва: Ленанд, 2021. - 104 с.

22. Цыганова, М. С. Разработка инструментария анализа данных о государственных закупках по федеральным законам № 44-ФЗ и № 223-ФЗ (на базе аналитической платформы Loginom) / М. С. Цыганова, С. В. Буреш, Д. А. Чернушенко // НЖ “Вестник Череповецкого Государственного университета”. – 2020. – №6. – С. 59–73.

Приложение А

Python-скрипт для вычисления показателей

```
import builtin_data
from builtin_data import InputTable, InputTables, InputVariables,
OutputTable, DataType, DataKind, UsageType
import pandas as pd
from datetime import datetime
from builtin_pandas_utils import to_data_frame,
prepare_compatible_table, fill_table

if InputTable:
    input_frame = to_data_frame(InputTable)

input_frame['count_uniq_prov_on_okved'] = 0 #количество уникальных
провайдеров по окведам
input_frame['count_part_prov'] = 0 #количество всех участий поставщика
по окведам
input_frame['count_uniq_okv_on_prov'] = 0 #количество уникальных
окведов, в которых участвововал поставщик
input_frame['count_win'] = 0 #количество побед
input_frame['count_win_on_okved'] = 0 #количество побед по окведам
input_frame['min_date_prov'] = 0 #минимальная дата участия у провайдера
input_frame['max_date_prov'] = 0 #максимальная дата участия у провайдера
input_frame['min_price_on_okved'] = 0 #минимальная стоимость лота по
окведам
input_frame['max_price_on_okved'] = 0 #максимальная стоимость лота по
окведам

input_frame = input_frame.sort_values(by=['okved_new','date',
'providerInn'], ascending=True)

for a in input_frame['okved_new'].unique():
    fr = input_frame[input_frame['okved_new'] == a]
    for x in fr['date'].unique():
        fr1 = fr[fr['date'] <= x]
        for c in list(fr1.index):
            if(input_frame.loc[c, 'count_uniq_prov_on_okved'] == 0):
                input_frame.loc[c, 'count_uniq_prov_on_okved'] =
len(fr1['providerInn'].unique())
for a in input_frame['providerInn'].unique():
    fr = input_frame[input_frame['providerInn'] == a]
    for c in list(fr.index):
        if(input_frame.loc[c, 'min_date_prov'] == 0):
            input_frame.loc[c, 'min_date_prov'] =
fr['publicationDateTime'].min()
        if(input_frame.loc[c, 'max_date_prov'] == 0):
            input_frame.loc[c, 'max_date_prov'] =
fr['publicationDateTime'].max()
    for x in fr['date'].unique():
```

```
        fr1 = fr[fr['date'] <= x]
        fr2 = fr1[fr1['winner'] == 1]
        for c in list(fr1.index):
            if(input_frame.loc[c, 'count_uniq_okv_on_prov'] == 0):
                input_frame.loc[c,      'count_uniq_okv_on_prov']      =
len(fr1['okved_new'].unique())
            if(input_frame.loc[c, 'count_part_prov'] == 0):
                input_frame.loc[c,         'count_part_prov']         =
len(fr1['okved_new'])
            if(input_frame.loc[c, 'count_win'] == 0):
                input_frame.loc[c, 'count_win'] = len(fr2['winner'])
            if(input_frame.loc[c, 'min_price_on_okved'] == 0):
                input_frame.loc[c,        'min_price_on_okved']        =
fr2['price_lot'].min()
            if(input_frame.loc[c, 'max_price_on_okved'] == 0):
                input_frame.loc[c,        'max_price_on_okved']        =
fr2['price_lot'].max()
input_frame['min_date_prov']                                           =
input_frame['min_date_prov'].astype('datetime64[ns]')
input_frame['max_date_prov']                                           =
input_frame['max_date_prov'].astype('datetime64[ns]')
output_frame = input_frame

# Если включена опция "Разрешить формировать выходные столбцы из кода",
структуру выходного набора можно подготовить по pd.DataFrame
if isinstance(OutputTable, builtin_data.ConfigurableOutputTableClass):
    prepare_compatible_table(OutputTable,                 output_frame,
with_index=False)
fill_table(OutputTable, output_frame, with_index=False)
```

Приложение Б

Python-скрипт для Random over-sampling

```
import builtin_data
from builtin_data import InputTable, InputTables, InputVariables,
OutputTable, DataType, DataKind, UsageType

import pandas as pd, numpy as np
from datetime import datetime
from builtin_pandas_utils import to_data_frame,
prepare_compatible_table, fill_table

if InputTable:
    input_frame = to_data_frame(InputTable)

count_class_0, count_class_1 = input_frame.part.value_counts()
df_class_0 = input_frame[input_frame['part'] == 0]
df_class_1 = input_frame[input_frame['part'] == 1]

df_class_1_over = df_class_1.sample(count_class_0, replace=True)
df_test_over = pd.concat([df_class_0, df_class_1_over], axis=0)

print(df_test_over.part.value_counts())

df_test_over['part'] = df_test_over['part'].astype('bool')
output_frame = df_test_over

if isinstance(OutputTable, builtin_data.ConfigurableOutputTableClass):
    prepare_compatible_table(OutputTable, output_frame,
with_index=False)
fill_table(OutputTable, output_frame, with_index=False)
```

Приложение В

Python-скрипт для Random under-sampling

```
import builtin_data
from builtin_data import InputTable, InputTables, InputVariables,
OutputTable, DataType, DataKind, UsageType

import pandas as pd, numpy as np
from datetime import datetime
from builtin_pandas_utils import to_data_frame,
prepare_compatible_table, fill_table

if InputTable:
    input_frame = to_data_frame(InputTable)

count_class_0, count_class_1 = input_frame.part.value_counts()

df_class_0 = input_frame[input_frame['part'] == 0]
df_class_1 = input_frame[input_frame['part'] == 1]
df_class_0_under = df_class_0.sample(count_class_1)
df_test_under = pd.concat([df_class_0_under, df_class_1], axis=0)

print(df_test_under.part.value_counts())

df_test_under['part'] = df_test_under['part'].astype('bool')
output_frame = df_test_under

if isinstance(OutputTable, builtin_data.ConfigurableOutputTableClass):
    prepare_compatible_table(OutputTable, output_frame,
with_index=False)
fill_table(OutputTable, output_frame, with_index=False)
```

Приложение Г

Python-скрипт Random OverSampler

```
import builtin_data
from builtin_data import InputTable, InputTables, InputVariables,
OutputTable, DataType, DataKind, UsageType

import pandas as pd, numpy as np
from datetime import datetime
from builtin_pandas_utils import to_data_frame,
prepare_compatible_table, fill_table
from imblearn.over_sampling import RandomOverSampler

if InputTable:
    input_frame = to_data_frame(InputTable)

labels = input_frame.columns[1:]
X = input_frame[labels]
y = input_frame['part']


X_ros, y_ros = RandomOverSampler().fit_resample(X, y)

fr = X_ros
fr['part'] = y_ros
fr['part'] = y_ros.astype('bool')
output_frame = fr

if isinstance(OutputTable, builtin_data.ConfigurableOutputTableClass):
    prepare_compatible_table(OutputTable, output_frame,
with_index=False)
fill_table(OutputTable, output_frame, with_index=False)
```