Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Удмуртский государственный университет»

Институт математики, информационных технологий и физики

Кафедра информационных систем и сетей

**Выпускная квалификационная работа на тему:**

«Разработка системы прогнозов тендерных закупок»

студента группы ОАБ-09.03.02-41

Комышев Д.А

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Организация руководства ВКР:

Ст. преподаватель Чалый И.А.

                                                                        Научный руководитель:

                                                                        к.т.н., доцент Дюгуров Д.В.

                                                                         \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

                                                                         Допущено к защите

                                                                         «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 года

                                                                         Заведующий кафедрой:

                                                                         к.т.н., доцент Дюгуров Д.В.

                                                                          \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ижевск, 2021

**Содержание**

[**Введение** 2](#_Toc75449692)

[**Глава 1. Теоретическая часть** 3](#_Toc75449693)

[**§1. Описание предметной области** 3](#_Toc75449694)

[**§2. Постановка задачи** 8](#_Toc75449695)

[**§3. Возможности моделей** 10](#_Toc75449696)

[**§4. Обзор используемых технологий.** 11](#_Toc75449697)

[**Глава 2. Реализация** 13](#_Toc75449698)

[**§1. Анализ полученных данных.** 13](#_Toc75449699)

[**§2. Построение и обучение моделей.** 16](#_Toc75449700)

[**Заключение** 35](#_Toc75449701)

[**Литература** 36](#_Toc75449702)

# **Введение**

В настоящее время существует огромная потребность бизнеса в том, чтобы знать, как закончится его будущая сделка, как со стороны потребителя, так и со стороны поставщика. Такая система помогла бы эффективнее подбирать партнера, продукт заказа или поставки, что в итоге благоприятно влияет на экономику всего рынка и страны.

Проект, созданный в рамках данной работы, представляет собой ряд моделей алгоритмов машинного обучения, способные прогнозировать те или иные исходы будущей сделки.

# **Глава 1. Теоретическая часть**

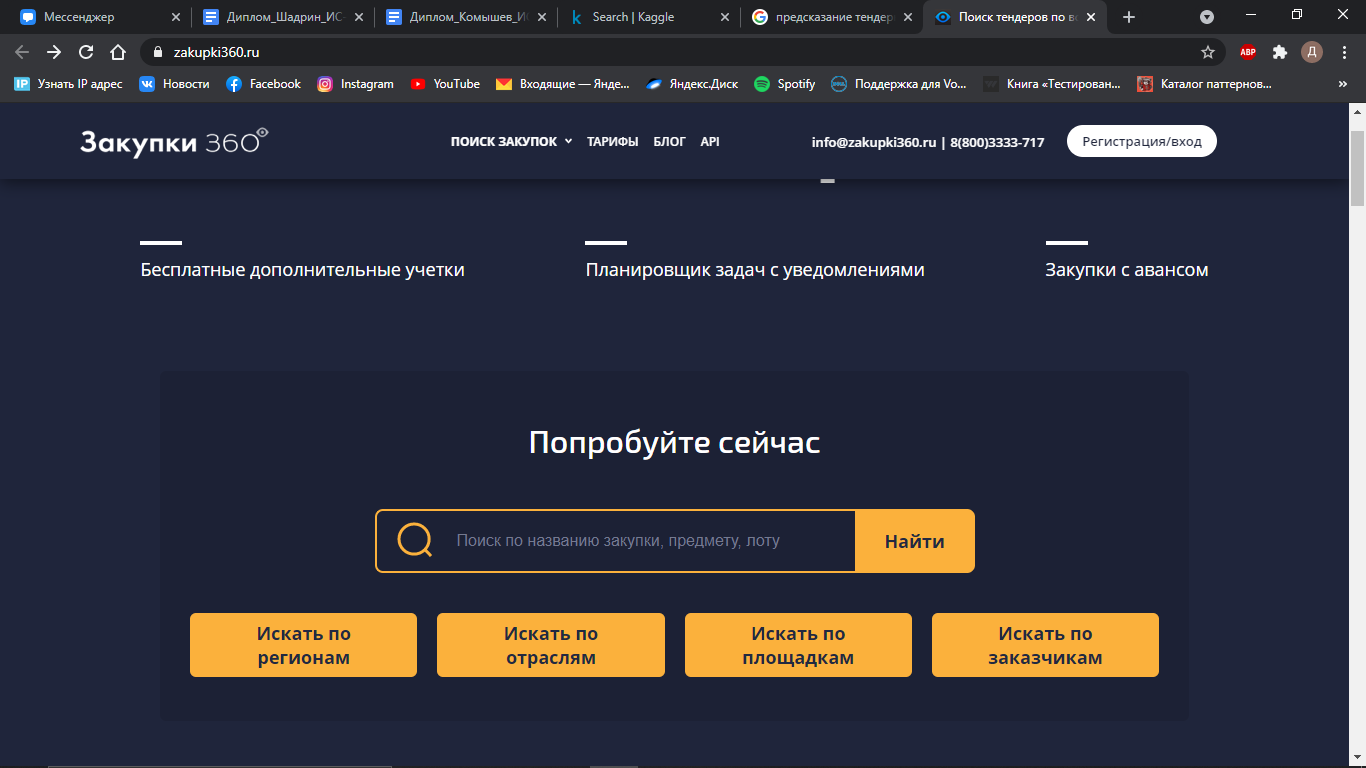
## **§1. Описание предметной области**

Цель данной работы – разработать обученные модели алгоритмов, после внедрения которых представители крупного и среднего бизнеса смогут лучше планировать свою деятельность.

Задачи:

1. Проанализировать рынок, сделать обзор существующих аналогов.
2. Изучить документацию, руководства и справочники, а также Интернет-ресурсы, связанные с тематикой работы.
3. Сбор данных для дальнейшего обучения моделей
4. Анализ данных, чтобы отсеять мешающие обучению данные
5. Разработка моделей
6. Обучение моделей
7. Выбор лучших моделей под каждую из задач
8. Внедрение моделей в сервис

Рассмотрим аналогичные продукты, чтобы понимать недостатки своего продукта и продуктов конкурентов.



*Рис.1 Сервис «Закупки 360»*

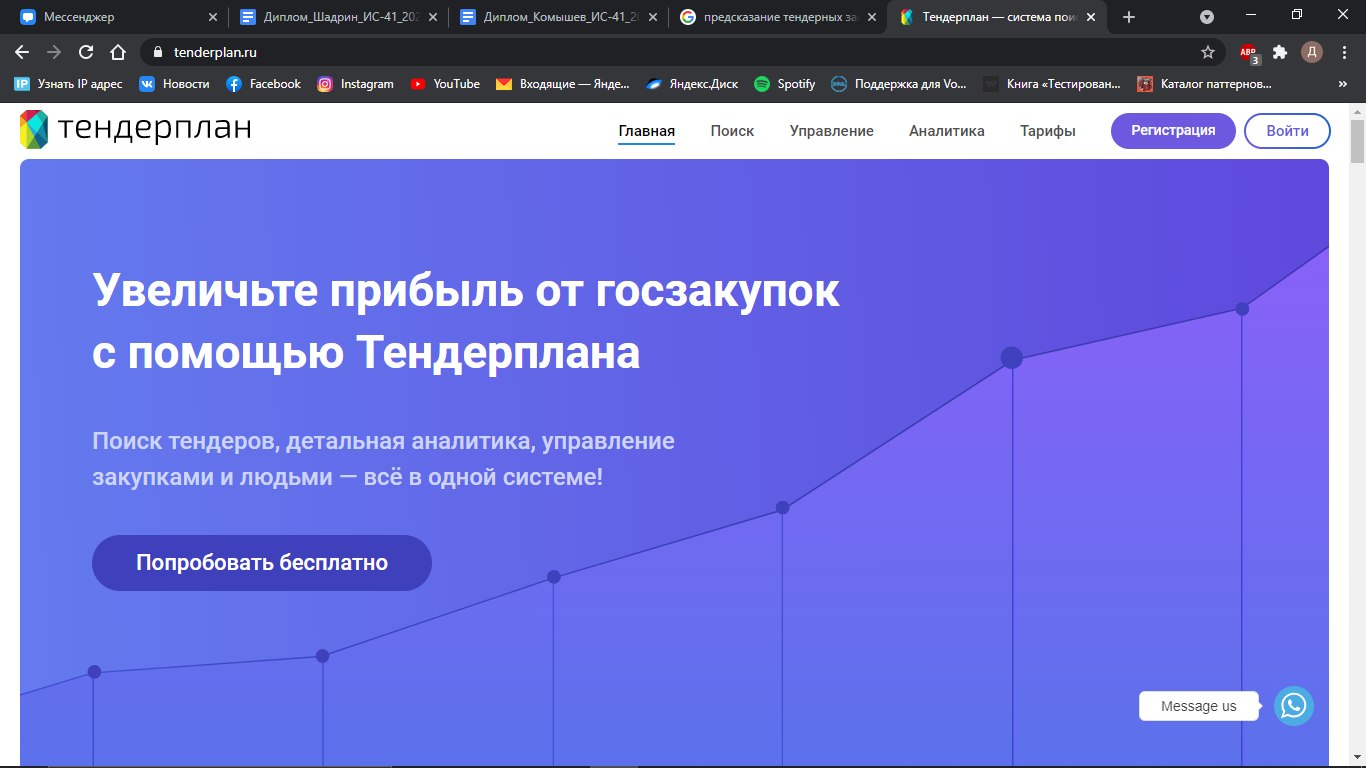
На рисунке 1 представлен сервис «Закупки 360».

Достоинства сервиса:

* Представляет бизнесу подходящие закупки
* Показывает вероятность участия конкурентов и их ценовые предложения
* Значительно сокращает время поиска подходящих контрактов

Недостатки сервиса:

* Скудный функционал бесплатной подписки
* Дорогая подписка на полноценный функционал



*Рис.2 Сервис «Тендерплан»*

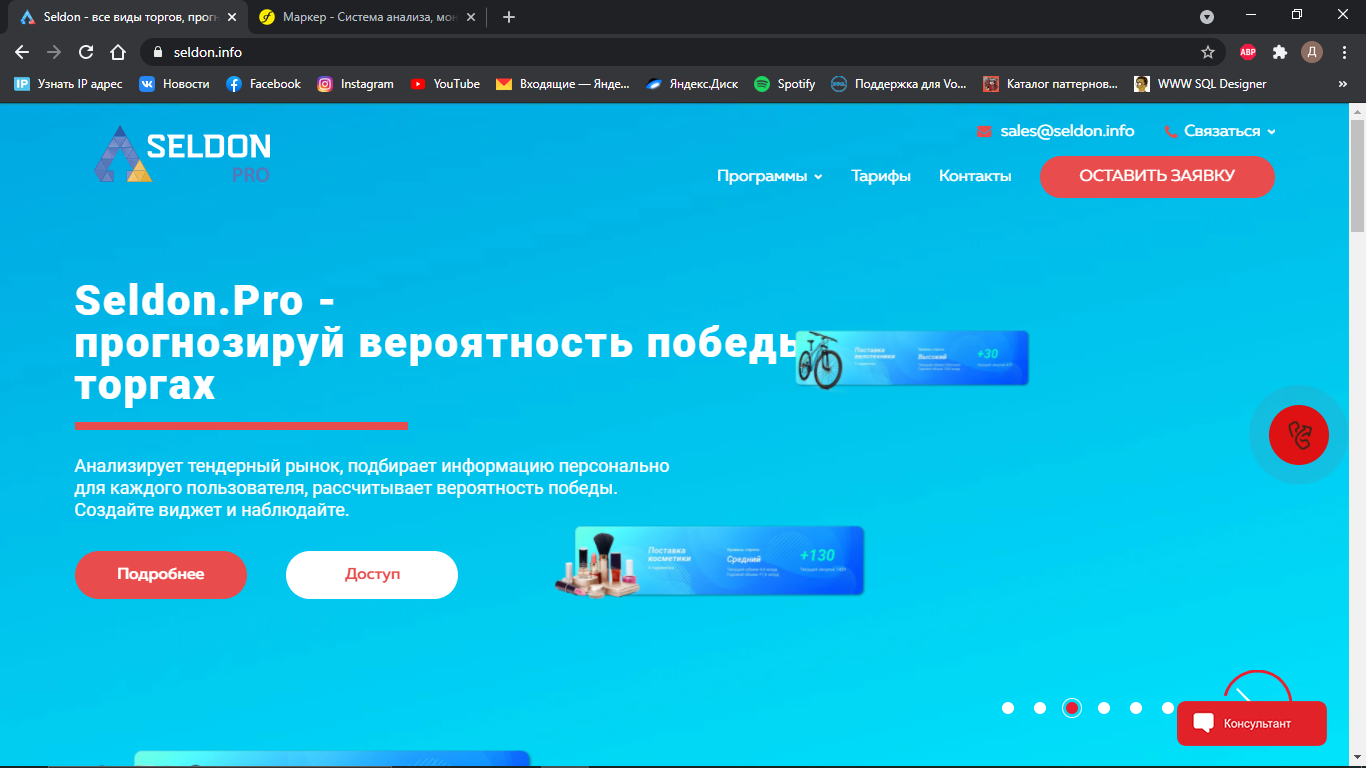
На рисунке 2 представлен сервис «Тендерплан».

Достоинства сервиса:

* Поиск надежных заказчиков
* Предсказание действий конкурентов
* Профилактика ненужных рисков и поиск новых направлений в бизнесе

Его недостатки:

* Аналогично с первым сервисом скудный бесплатный функционал
* Дорогая подписка с доступом ко всем функциям



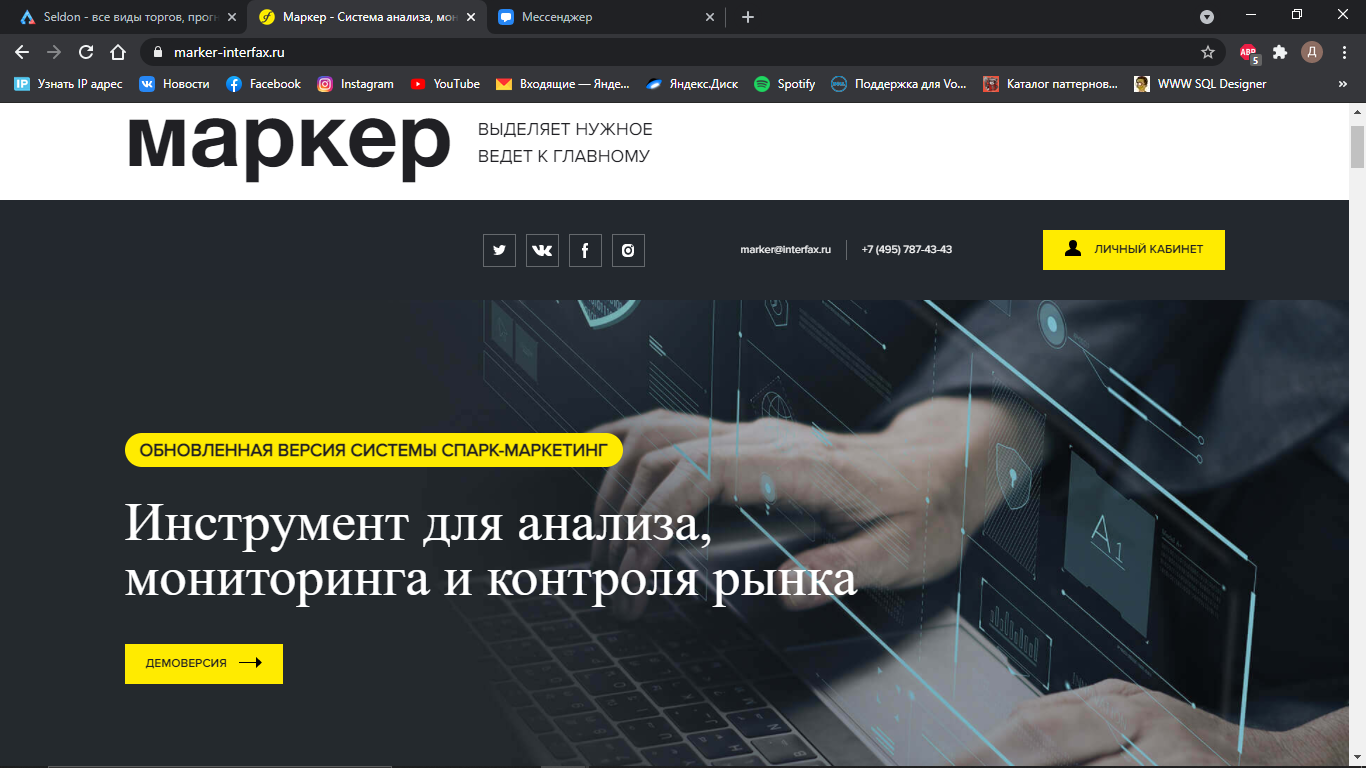
*Рис.3 Сервис «Seldon»*

На рисунке 3 представлен сервис «Seldon». Он имеет следующие преимущества:

* Прогнозирование конкурентов, получение данных о потенциальных поставщиках
* Рейтинг поставщиков, заказчиков
* Информация обо всех видах торгов
* Анализ в реальном времени

Недостатки:

* Дорогая подписка



*Рис. 4. Сервис «Маркер»*

На рисунке 4 представлен сервис «Маркер». Преимущества сервиса:

* Персональный мониторинг новых закупок
* Выявление признаков нарушения в закупках
* Анализ рынка и его участников
* Динамика цен на товары и услуги
* Себестоимость участия по своей номенклатуре
* Факторы риска
* Статистика по участию фирм-конкурентов в закупках

Недостатки:

* Дорогая подписка

## **§2. Постановка задачи**

Проект представляет собой ряд алгоритмов машинного обучения (такие как нейронная сеть, градиентные бустинги, логическая и линейная регрессии и так далее), предсказывающие, как хорошо закончится тендерная закупка для потребителя и поставщика.

Для успешной реализации необходимо решить ряд вопросов:

* Сбор данных с официального сайта тендерных закупок
* Анализ собранных данных
* Разработка алгоритмов машинного обучения
* Обучение алгоритмов
* Сопоставление результатов и выбор наилучшей
* Внедрение моделей в сервис

Данные берутся с государственного сайта госзакупок zakupki.gov. Далее происходит визуальный и числовой анализ полученных данных (графики отношений двух параметров, диаграммы, тепловые карты, матрицы корреляции и так далее). Анализ необходим, чтобы понять, есть ли корреляция между данными, существуют ли выбросы, от которых стоит избавиться. Для это сначала предлагается некая гипотеза, например, что существует корреляция между ценой и временем исполнения, дальше создаются графики, которые подтверждают или опровергают эту гипотезу. После проверки всех гипотез и обработки данных (обработка основывается на анализе, например, удаляются выбросы, которые портят статистику) предлагаются алгоритмы машинного обучения, подходящие для данной задачи. Дальше прогоняются ряд алгоритмов и отдается предпочтение алгоритму с лучшим результатом для этой задачи. Уже после всей проделанной работы обученная модель алгоритма передается разработчику сервиса, который внедряет модель в сервис.

Из проблем, которые могут возникнуть в процессе разработки, это малая корреляция между параметрами, например, что время или тип заказчика не очень зависят от цены, отчего точность предсказания цены будет низкой. Также технические трудности, например, слабая видеокарта, которая необходима для тензорных операций, в частности обучения алгоритмов, например, аппроксимация весов нейронной сети.

## **§3. Возможности моделей**

Проект представляет собой ряд обученных моделей машинного обучения, которые могут предсказывать следующие моменты:

* Предсказание будущей цены на исполнение тендерного контракта
* Пройдет ли заявка на тендерный аукцион
* Закончатся ли деловые отношения заказчика с поставщиком успешно

## **§4. Обзор используемых технологий.**

**Pandas**

Это программная библиотека, написанная для языка программирования Python, которая может читать .csv, .txt и другие файлы, содержащие структурированные данные, и хранить эти данные в собственной структуре данных, схожей со словарем, и выводить в виде таблицы.

**NumPy**

Это программная библиотека, имеющая функционал операций линейной алгебры, например инициализация матриц или векторов, их умножение и другие алгебраические операции, такие как транспонирование и так далее.

**Seaborn**

Программная библиотека, написанная для языка Python, помогающая программистам визуализировать данные, построить графики или диаграммы на их основе.

**Matplotlib**

Можно сказать, что он конкурент библиотеке Seaborn, он аналогично создает интерактивные визуализации данных.

**Keras**

Является интерфейсом для программной библиотеки TensorFlow, которая в свою очередь предназначена для инициализации и обучения искусственной нейронной сети.

**Scikit-learn**

Программный набор инструментов для Data Science. В нем имеется такой функционал, как набор метрик точности для моделей регрессии или классификации, деревья принятия решения, градиентные бустинги, например, AdaBoost, инструмент грубого подбора параметров для моделей GridSearchCV и многое другое.

**Catboost**

Библиотека от компании Яндекс, которая предоставляет алгоритм собственной разработки, основанный на ансамбле деревьев решения и градиентом спуске. Главная отличительная особенность от других подобных алгоритмов – это уникальная внутренняя обработка категориальных данных.

**XGBoost**

Конкурент библиотеки Catboost, который тоже берет за основу алгоритм ансамбля деревьев решения и градиентного спуска. Один из первых в топе алгоритмов машинного обучения по уровню предсказания.

# **Глава 2. Реализация**

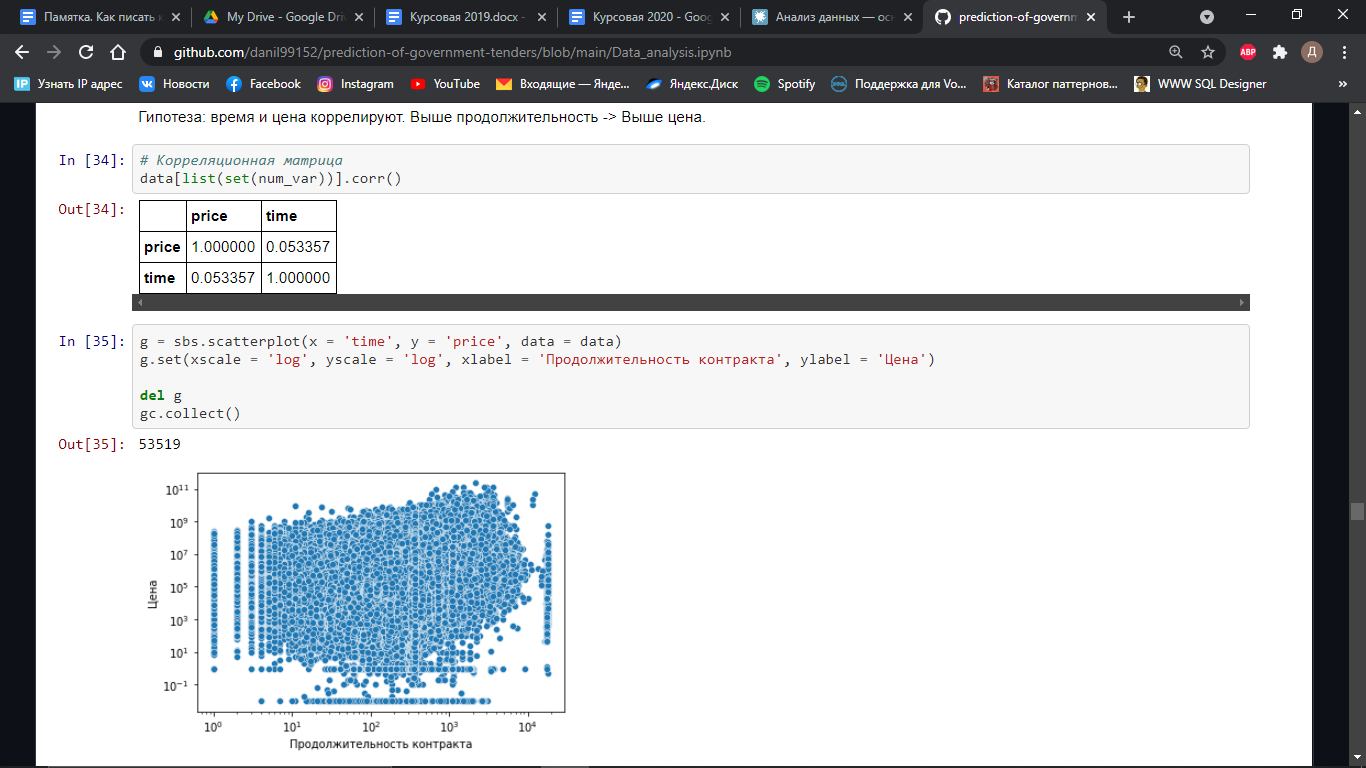
Проект представляет собой визуальный анализ данных, описывающие тендерные закупки, и набор обученных моделей алгоритмов машинного обучения, которые могут предсказывать разные состояния этих самых тендерных закупок. Поэтому глава реализации разделится на две части.

## **§1. Анализ полученных данных.**

Роль анализа данных заключается в том, чтобы отсеять лишнюю информацию, проверить все возможные взаимосвязи между данными и на основе этого в дальнейшем строить алгоритмы обучения.

Для начала предлагается некая гипотеза, потом осуществляется проверка этой гипотезы, а дальше вывод, в котором записывается, подтвердилась ли гипотеза или нет и почему. Рассмотрим пример.

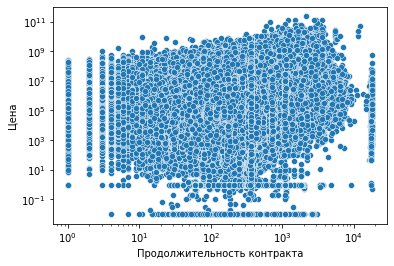
Есть гипотеза, что цена и время на прямую влияют друг на друга: выше продолжительность исполнения контракта – выше цена контракта. Может показаться, что это очевидно, но в некоторых случаях такая мысль ошибочна, поэтому желательно проверить те вещи, которые напрямую влияют на точность будущей модели.



*Рис.5. Корреляционная матрица.*

На рисунке 5 представлена матрица корреляции параметров времени исполнения и цены. На ней можно заметить, что влияние двух параметров друг на друга не велико: около 5% друг от друга. Поэтому стоит рассмотреть поподробнее.

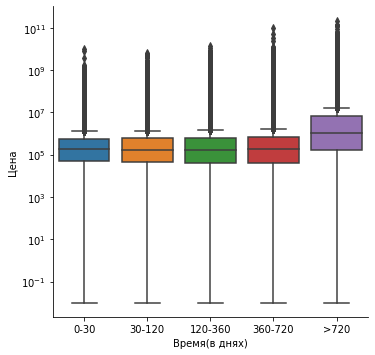
Попробуем вывести диаграмму рассеяния контрактов по цене и времени.



*Рис.6. Диаграмма рассеяния контрактов.*

С помощью библиотеки Seaborn получилось вывести диаграмму, которая показана на рисунке 6. На ней можно увидеть небольшие выбросы в правом верхнем углу.

Было решено разбить время на периоды: 0–30 дней, 30–120 дней, 120–360 дней, 360–720 дней и больше 720 дней. Каждый из периодов вывести в качестве ящика с усами или диаграммы размаха. Этот график удобен тем, что он выявляет медиану – линия в середине ящика, границы ящика – первый и третий квартили, а концы усов – края статистически значимой выборки. Данные, выходящие за границы усов в виде точек – выбросы – это те значения, которые выделяются из общей выборки, например контракт за несколько миллиардов, когда в общей массе нет миллиардных контрактов.



*Рис. 7. Диаграмма размаха временных диапазонов от цены.*

На рисунке 7 заметно, что контракты с временным периодом более 720 дней имеют тенденцию на рост цены. По большей части они и создают корреляцию в 5%.

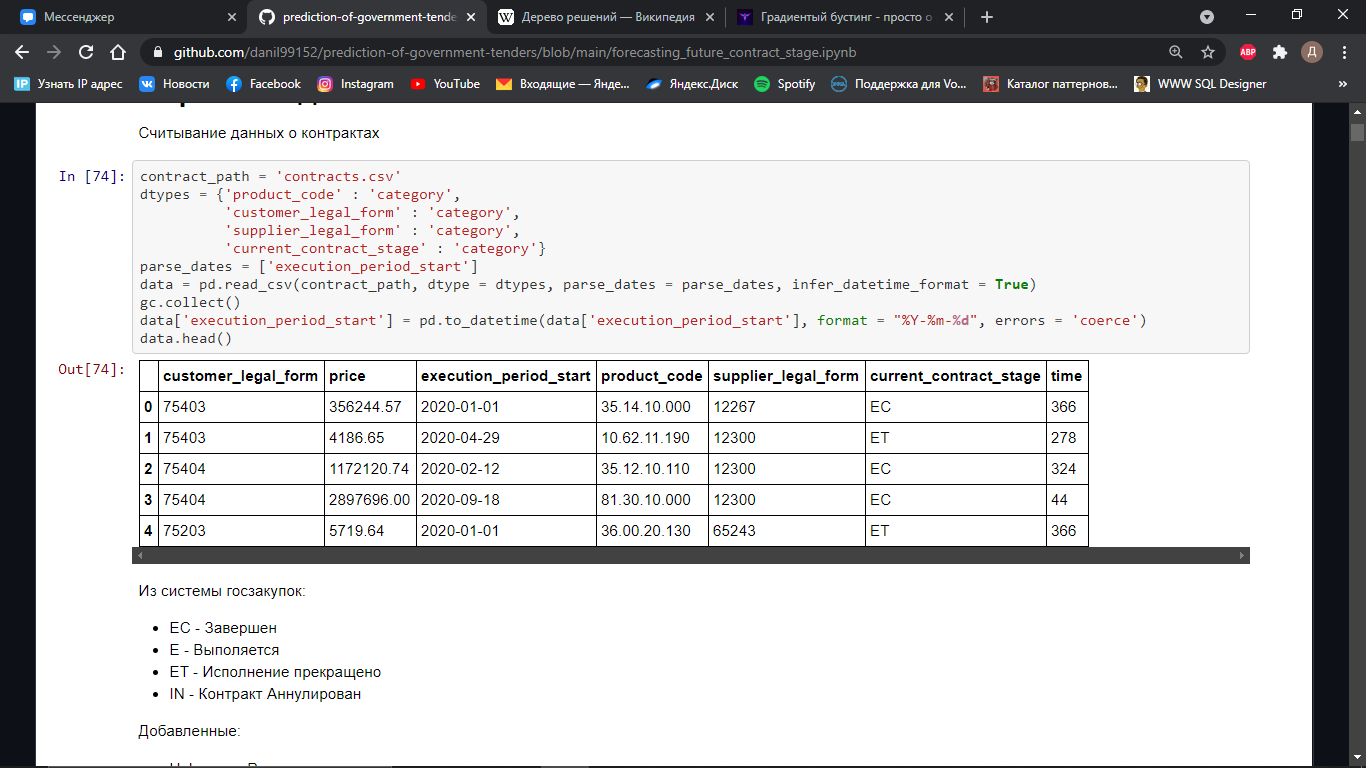
После анализа зависимости продолжительности исполнения контракта от цены записывается вывод, в котором подтверждается гипотеза, но с оговоркой, что корреляция присутствует для контрактов с продолжительностью более 720 дней.

## **§2. Построение и обучение моделей.**

После анализа всех зависимостей между параметрами, очистки от лишних выбросов, подтверждения или опровержения гипотез начинается самое главное – построение и обучение предсказательных алгоритмов.

Начнем с примера построения и обучения алгоритмов, нацеленных на предсказание завершенности тендера, то есть расторжением контракта или успешным его исполнением.

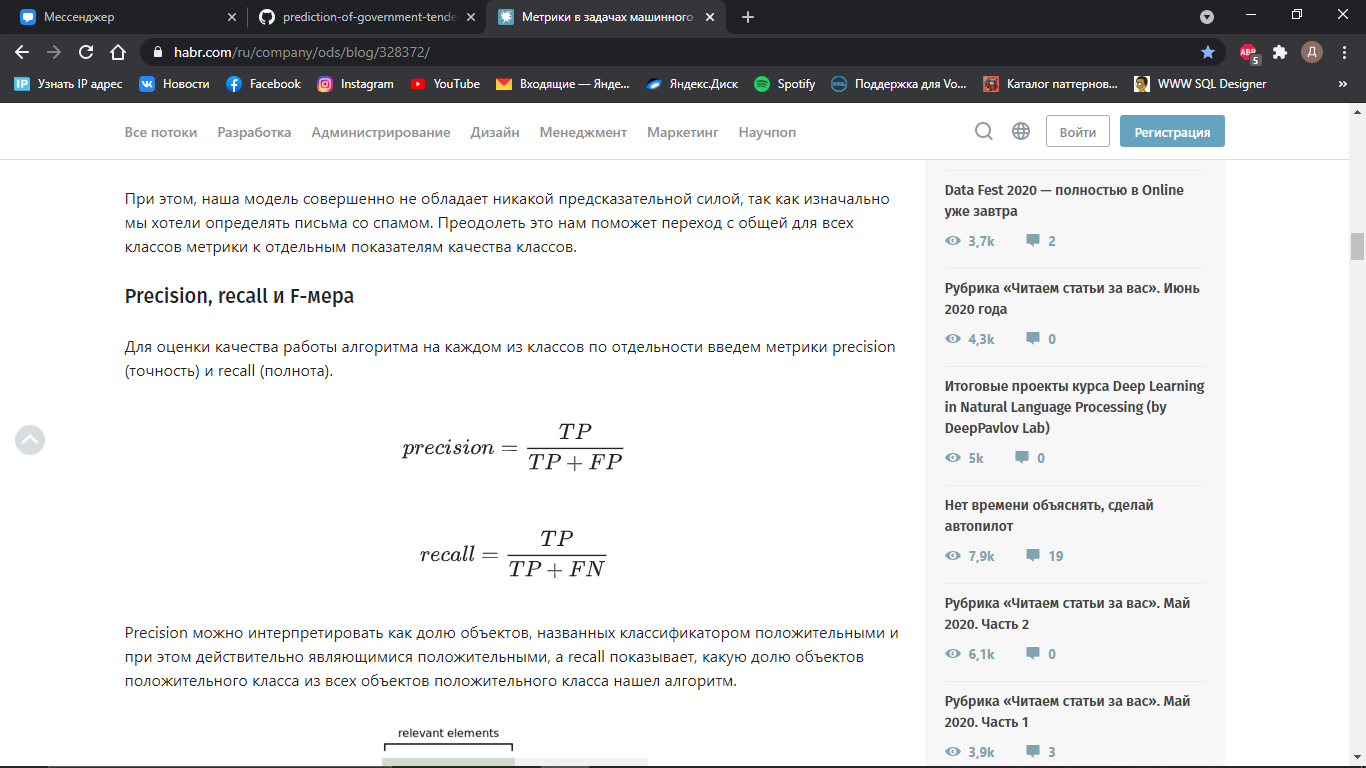
Пример данных, которые участвовали в обучении алгоритма представлен на рисунке 8.



*Рис.8. Пример данных в виде таблицы Pandas.*

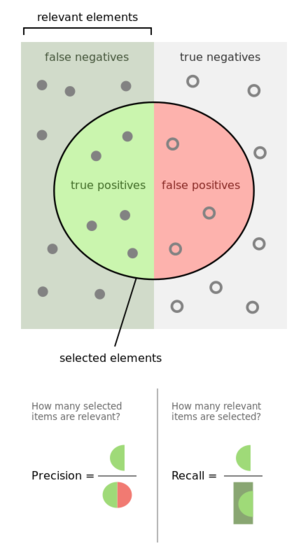
За входные данные были взяты все столбцы, кроме current\_contract\_stage, эти данные пошли в качестве выходных. В current\_contract\_stage всего два класса: EC (execution canceled) и ET (execution terminated). Бизнес-задача заключалась в том, чтобы негативный исход, а именно класс ET, прогнозировался с полнотой около 90%. Для этого было решено оставить тестовые данные (20% от всего набора данных) в исходном виде, при этом в нем содержались новейшие данные, а более старые (остальные 80%) было решено сбалансировать в таком виде, чтобы алгоритм предсказывал полноту нужного класса на кросс-валидации в требуемом виде. В итоге отношение классов EC к ET равнялось 25%, а на тестовых данных отношение ET к EC – 19%. Этот же метод применялся и в следующей задаче, где полнота класса True на кросс-валидации с реальными новейшими данными должна быть также уровне 90%.

Precision – точность, доля объектов, названные классификатором как положительные и при этом являющиеся положительными, а recall – полнота, доля объектов положительного класса из всех объектов положительного класса, который нашел алгоритм.



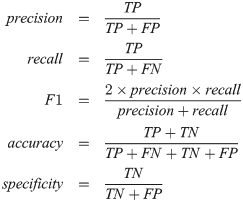
*Рис.9. Формула точности и полноты предсказания объектов класса.*

На рисунке 9 представлены формулы точности (precision) и полноты (recall), где TP – true positive (положительный), FP – false positive (ложноположительный), FN – false negative (распознан как принадлежащий к другому классу, но в действительности принадлежит этому). Более подробно эти переменные объясняются далее в таблице 1.



*Рис.10. Визуализация двух метрик.*

Метрика f1 – это золотая середина между полнотой и точность и выражается в формуле, которая указана на рисунке 11, где precision – точность, recall – полнота.



*Рис.11. Формула метрики f1.*

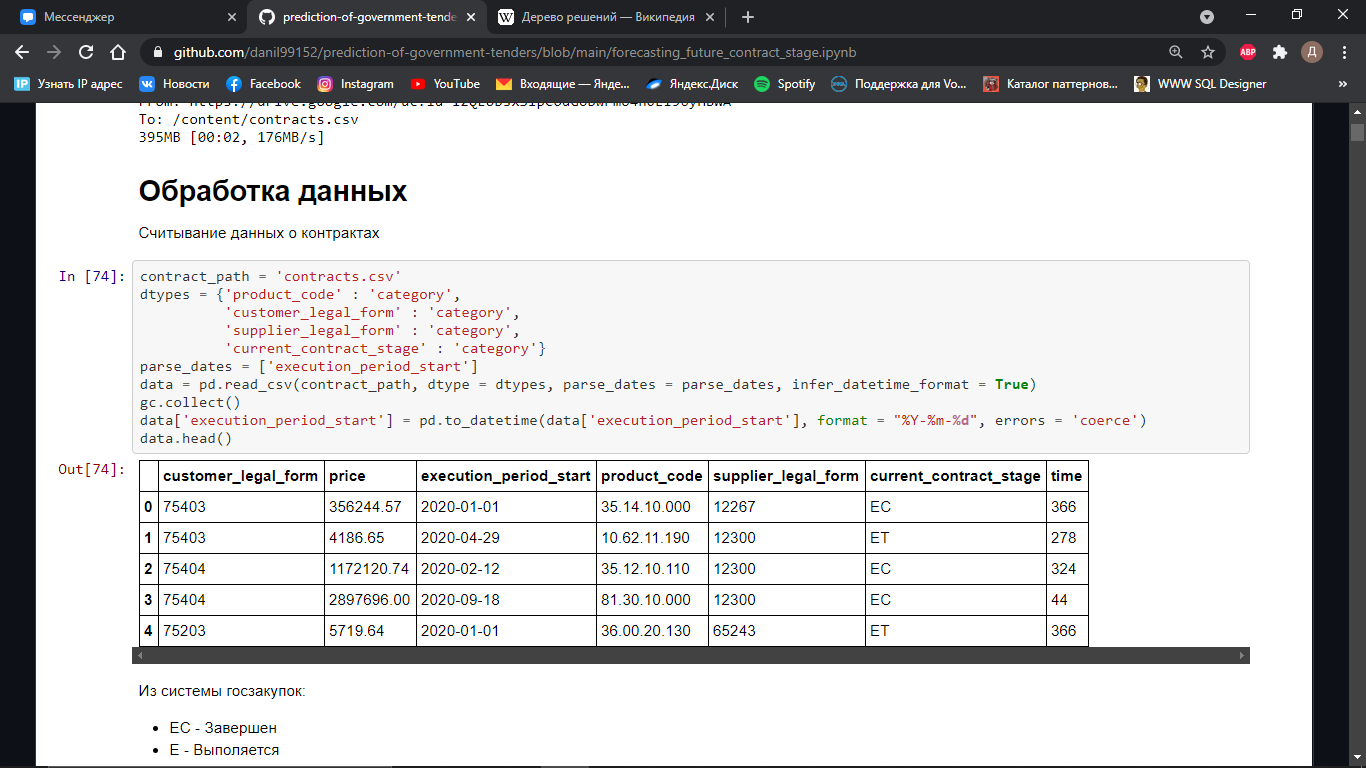
Параметр «support» просто указывает, сколько объектов участвовало в оценке качества модели для каждого класса.

Проведен эксперимент с разными алгоритмами: от простых типа линейной регрессии до классических нейронных сетей. Взор пал на алгоритм градиентного бустинга (GBM). Немного о самом алгоритме. Это сложный алгоритм построения ансамбля из малых предсказателей, обычно деревьев принятия решения, для задач регрессии или классификация. Идея заключается в том, что малые предсказатели построены последовательно друг от друга и следующая модель учится на ошибках предыдущей. Классические вариации этого алгоритма – это XGBoost, CatBoost, LightGBM.

Рассмотрим пример построения алгоритма CatBoost и дальнейшего его обучения.

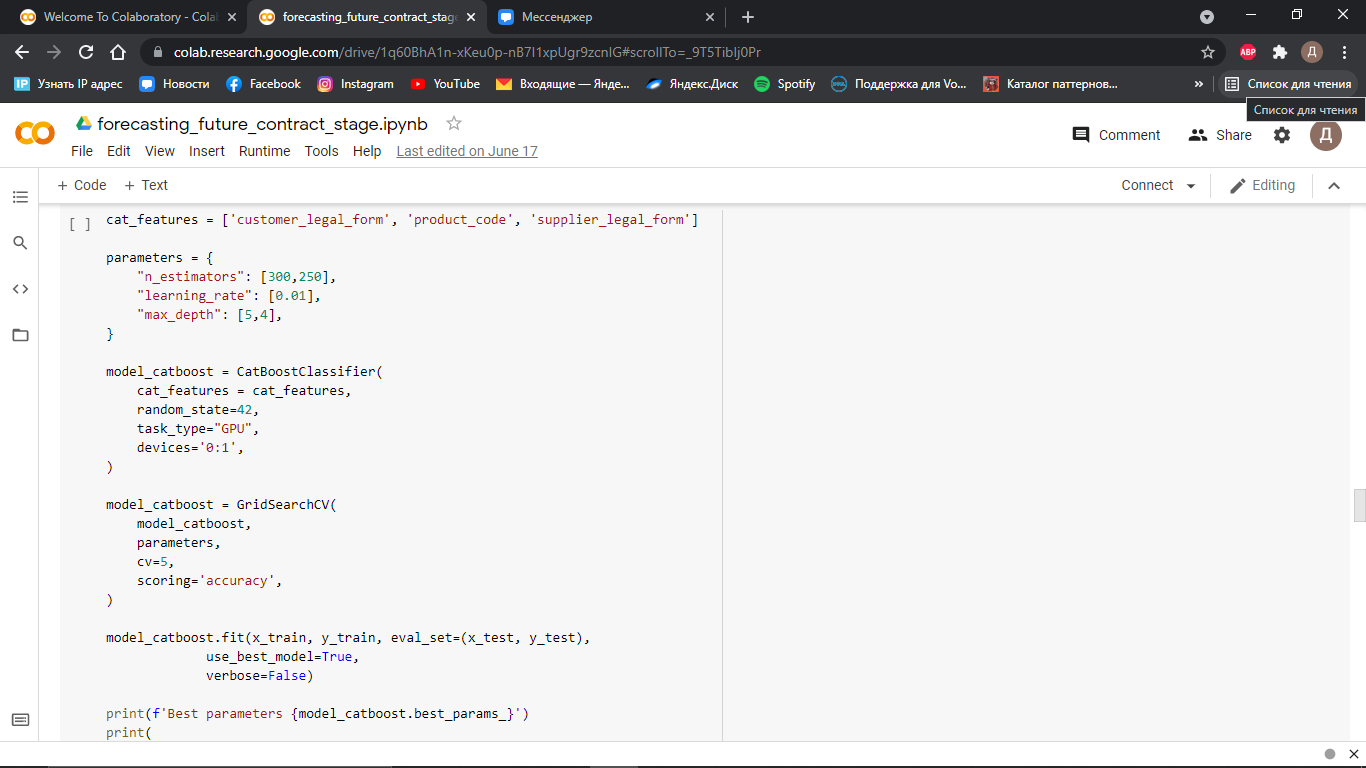
Для начала необходимо импортировать нужную библиотеку, это делается с помощью одной строчки на языке Python: from catboost import CatBoostClassifier. Класс CatBoostClassifier отвечает за инициализацию самого алгоритма.

Обычно перед тем, как скормить данные алгоритму, надо их подготовить, алгоритмы на основе ансамблей деревьев принятия решения хороши тем, что не нуждаются в нормализации или стандартизации данных, поэтому только присвоим некоторым данным категориальный тип:



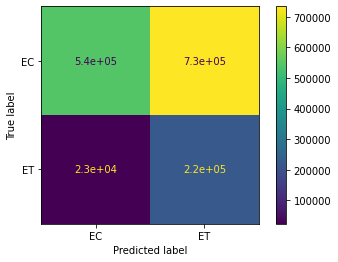
*Рис. 12. Обработка данных.*

CatBoost имеет преимущество в том, что работает с категориальными данными напрямую без перевода в число (Label Encoding) или числовой вектор (One-Hot Encoding). Достаточно только указать в списке категориальные параметры и вызвать метод fit вместе с ним.



*Рис.13. Инициализация модели алгоритма CatBoost.*

На рисунке 13 видно, что параметры модели перебираются грубым методом с помощью библиотеки GridSearchCV в указаном диапазоне, а также обучение на GPU и выбор номера устройства GPU.



*Рис.14. Матрица ошибок.*

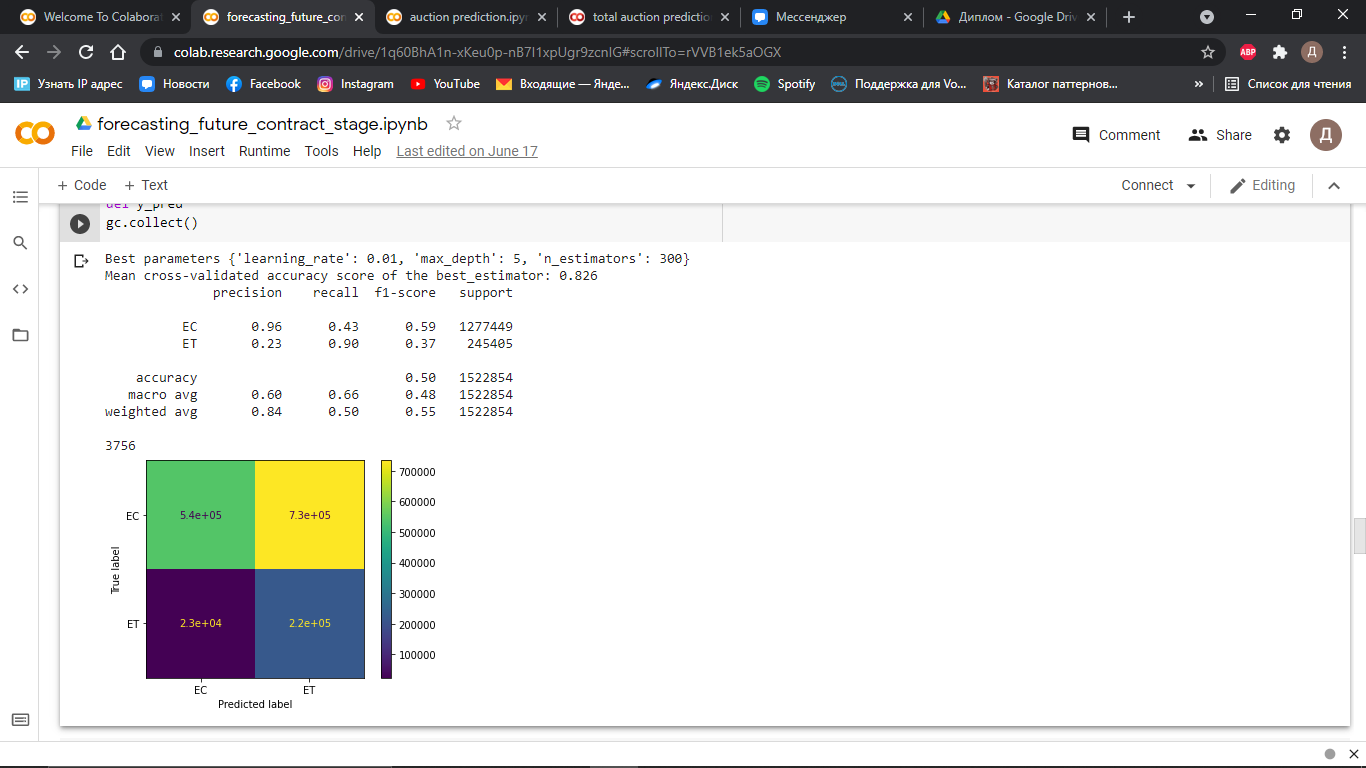
На рисунке 14 представлена матрица ошибок, в которой указано, какое количество объектов модель присвоила к каждому из классов и какое количество на самом деле принадлежит.

Читать матрицу нужно следующим образом.

|  |  |  |
| --- | --- | --- |
|  | y=1 | y=0 |
| =1 | Достоверно Положительный (TP) | Ошибочно Положительный (FP). |
| =0 | Ошибочно Отрицательный (FN) | Достоверно Отрицательный (TN) |

**Таблица 1. Описание матрицы ошибок.**

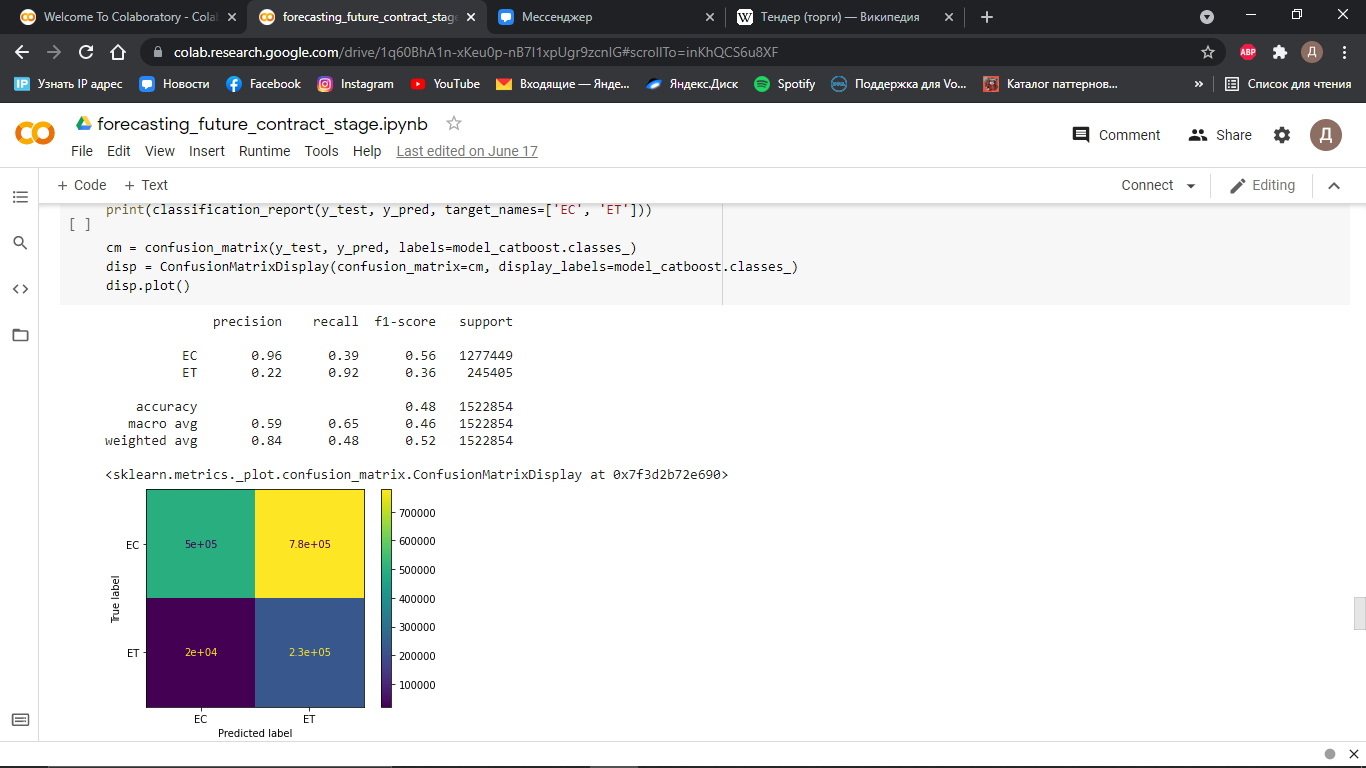
Здесь  — это присвоенный объект классу, а y — объект, на самом деле принадлежащий классу.  
Таким образом, ошибки классификации бывают двух видов: Ошибочно Отрицательный (FN) и Ошибочно Положительный (FP).



*Рис.15. Результат качества модели на тестовом множестве данных.*

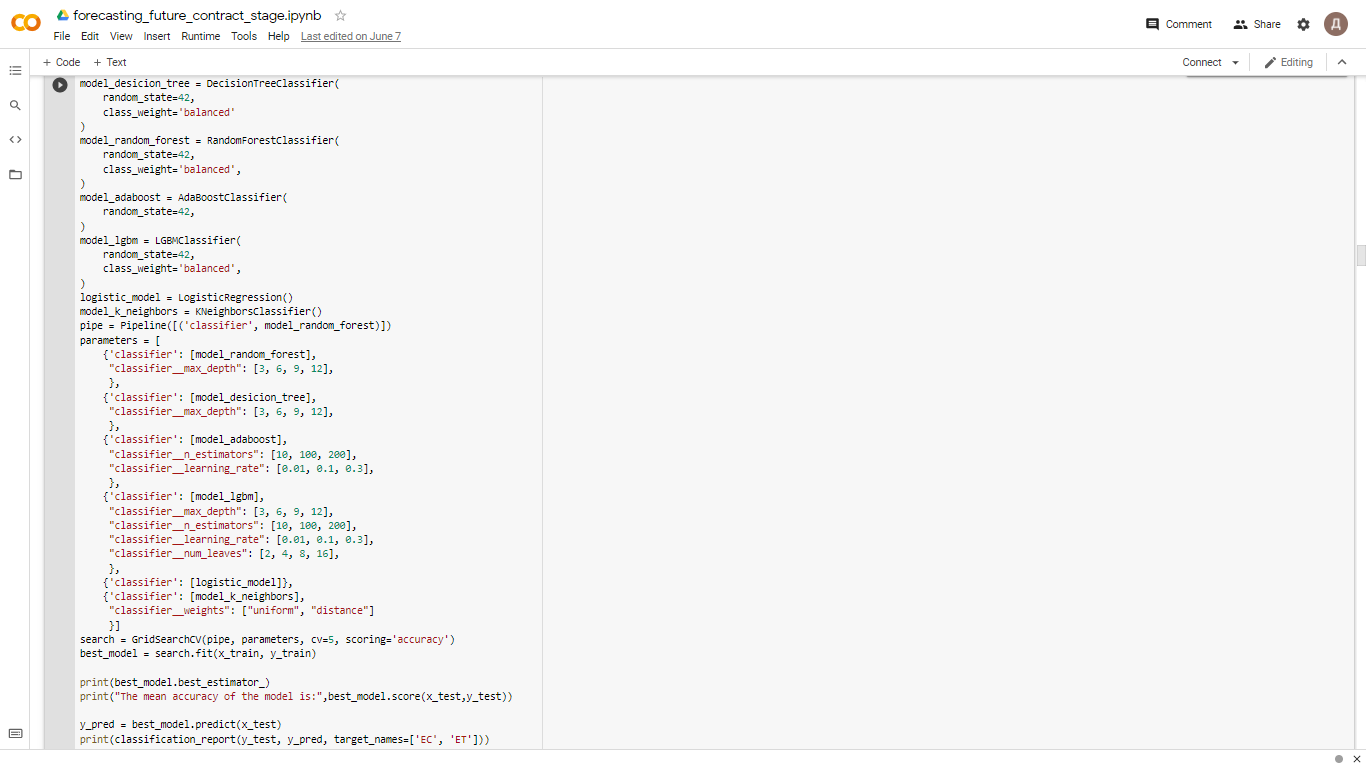
На рисунке 15 указано качество предсказания модели для каждого класса, а также лучшие подобранные параметры для модели, основываясь на скорости обучения и результате кросс-валидации.

Можно заметить, что тут есть такой параметр как supplier\_legal\_form (форма поставщика), о котором не будет известно на момент предсказания, а именно когда клиент захочет узнать, как завершится его тендерный контракт, до начала аукциона. Эта модель была сделана для поставщиков, чтобы понять риск участия в этой тендерной заявке. Для клиентов же тендерной заявки была написана отдельная модель без этого параметра. Соответственно балансировка между двумя классами тоже была уникальна: отношение EC к ET равно 37%. Результат этой модели на рисунке 16.



*Рис.16. Результат качества модели на тестовом множестве данных для второй модели.*

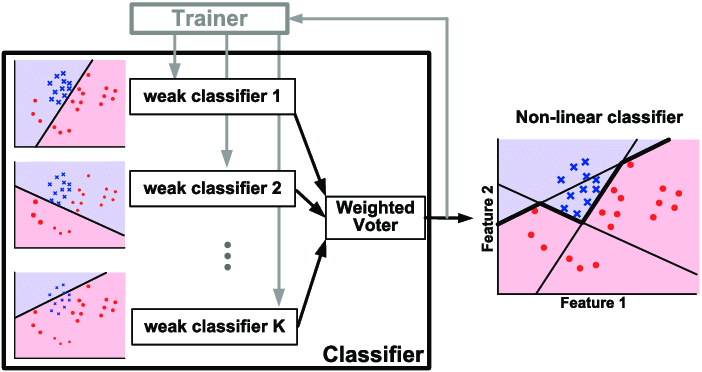
Интересно сравнить результат кросс-валидации CatBoost с другими алгоритмами. Для этого есть в библиотеке scikit-learn инструмент Pipline. В связке с GridSearchCV он сделает перебор всех алгоритмов со всеми заданными параметрами. Важно понимать, что сравнение алгоритмов проводилось на другом количестве и балансировке тренировочных и тестовых данных. Этот этап позволил выделить алгоритм CatBoost в победители для дальнейшего его использования. Пример реализации метода Pipline на рисунке 17.



*Рис.17. Реализация поиска лучшей модели с подходящими параметрами.*

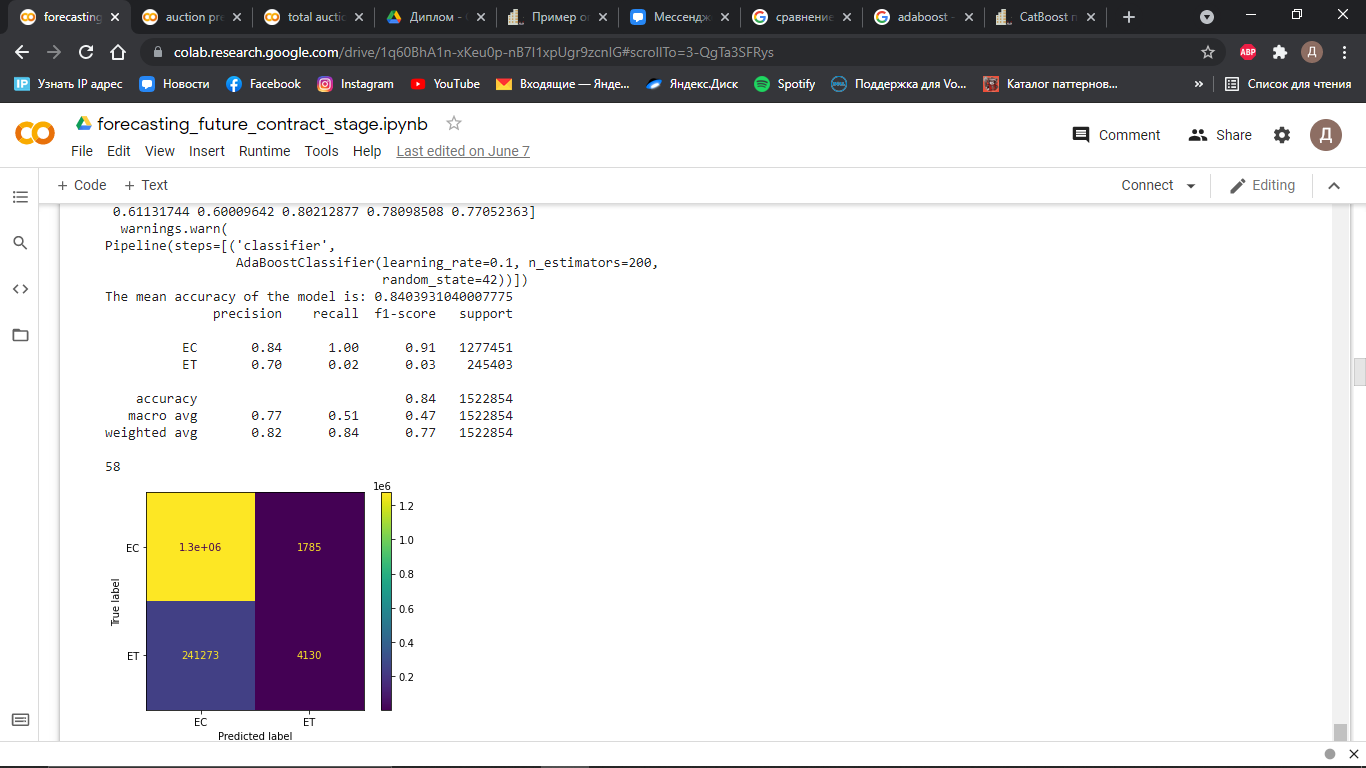
На рисунке 17 можно заметить, что поиск шел по таким алгоритмам, как дерево решений, лес деревьев решения, Adaboost, LightGBM, логистическая регрессия и алгоритм кластеризации метод k-средних.

К сожалению, LightGBM проиграл, так как он является единственным из рода алгоритмов градиентных бустингов в этом списке, а победил AdaBoost. AdaBoost является алгоритмом адаптивного бустинга, главное свойство адаптивных бустингов в том, что он объединяет слабые классификаторы (деревья решений) в «комитет», каждый следующий комитет классификаторов строится по объектам, неверно классифицированным предыдущими комитетами. Пример обучения адаптивного бустинга на рисунке 18.



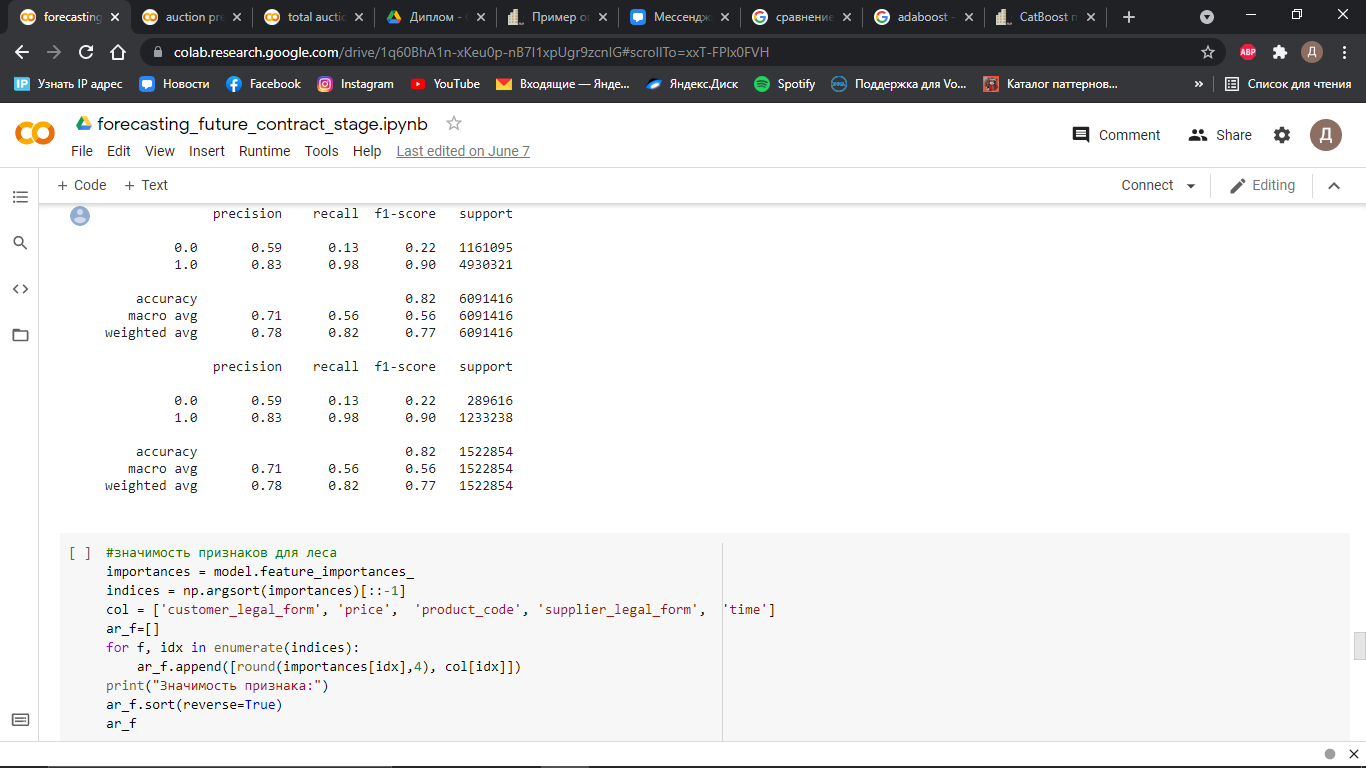
*Рис.18. Обучение AdaBoost.*

Результат кросс-валидации лучшей модели на рисунке 19.



*Рис.19. Результат кросс-валидации AdaBoost.*

Так как адаптивный бустинг больше подвержен выбросам и шумам в данных, а также не умеет обрабатывать категориальные данные, как CatBoost или LightGBM, пришлось отказаться от него в дальнейшей разработке. Также XGBoost показал еще лучше результат, чем AdaBoost при той же балансировке данных. Результат на рисунке 20.

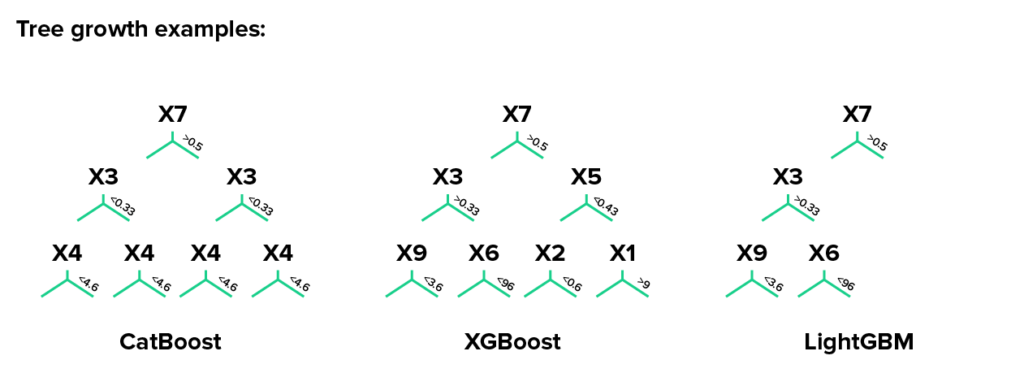


*Рис.20. Результат XGBoost.*

На рисунке нулевым классом обозначен класс ET, первый – EC. При той же балансировке CatBoost показывал лучший результат из всех алгоритмов. Позже пришлось остановиться на нем и на следующих задачах. Последний его результат для этой задачи в начале главы.

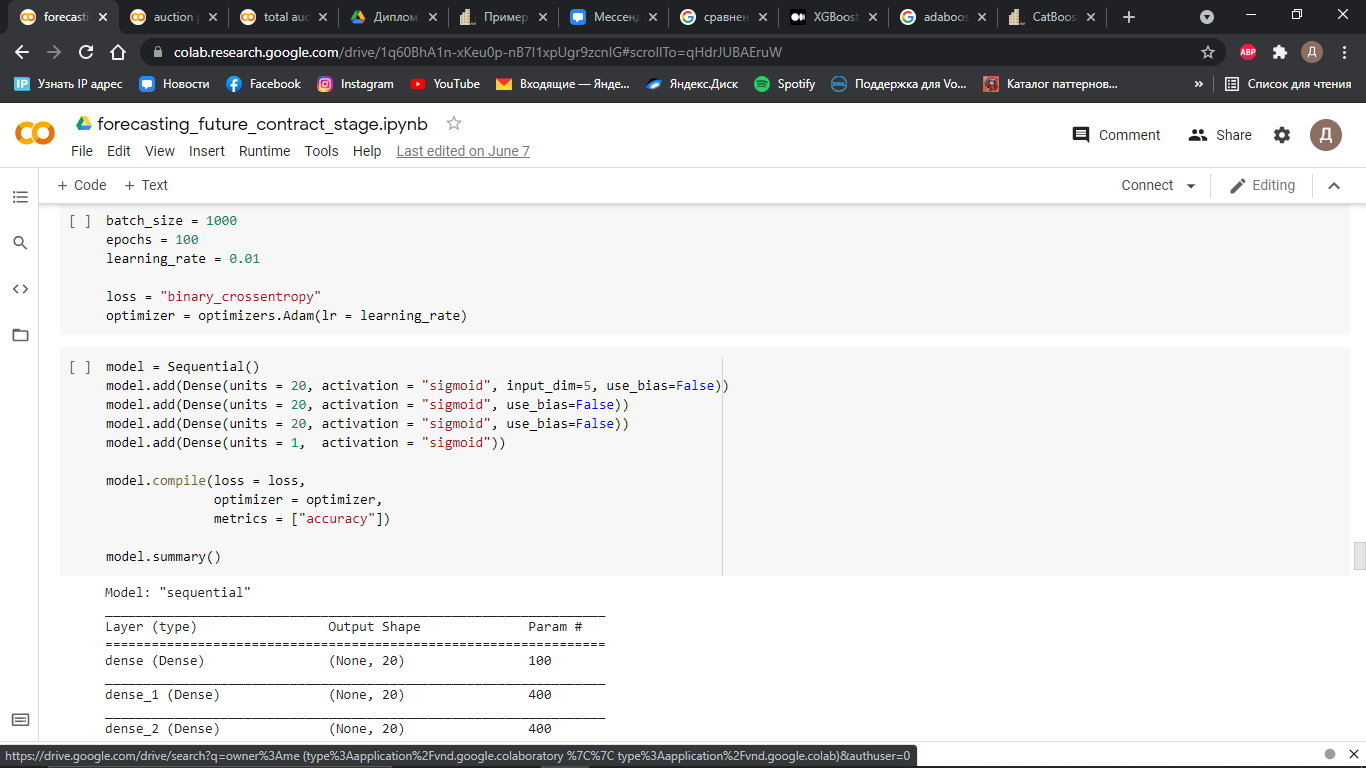
CatBoost экспериментально подошел лучше всего для конкретного типа задач, рассмотрим почему так вышло:

1. Эффективный способ кодирования категориальных данных без применения метода LabelEncoder (обычный перевод в числовое значение, что приводит к переобучению).
2. Симметричные деревья, что приводит к наименьшей ошибке (рисунок 21, сравнение деревьев).



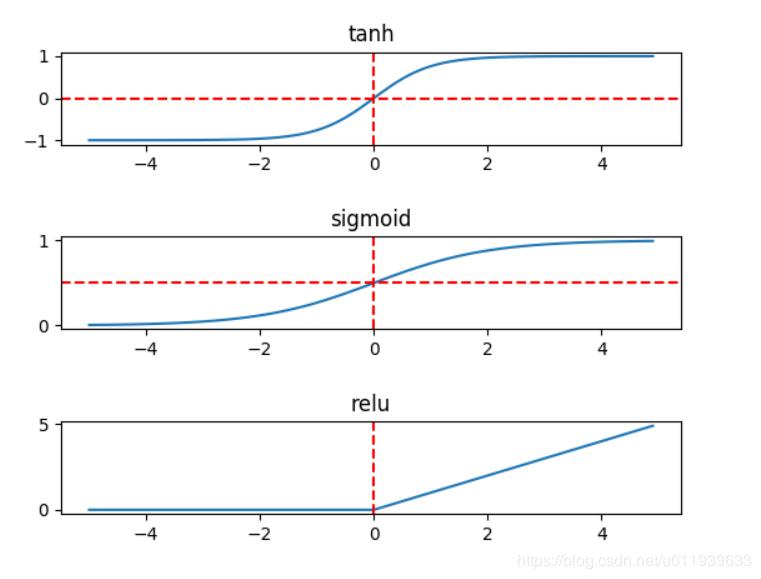
*Рис.21. Сравнение структур деревьев.*

Также в качестве классификатора в эксперименте участвовала полносвязная нейронная сеть или перцептрон. На рисунке 22 модель с лучшим результатом кросс-валидации среди других вариантов модели нейронной сети.



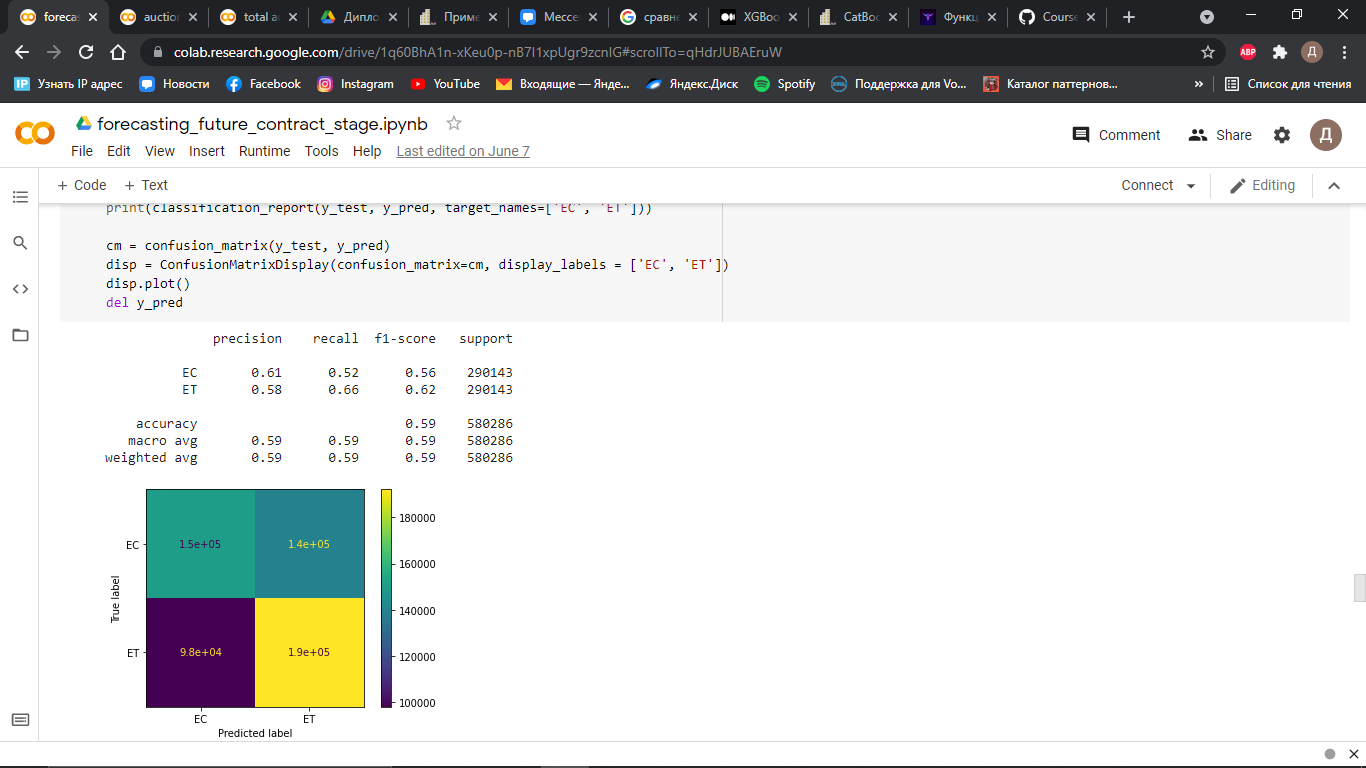
*Рис.22. Модель нейронной сети.*

Количество слоев и нейронов было подобрано интуитивно и экспериментально, функция сигмоиды была выбрана благодаря лучшей обучаемости, по сравнению с ReLU, так как у второй для отрицательных входных значений часть функции представляет горизонтальную линию, что приводит к невозможности аппроксимации весов и пассивности нейронов, потому что градиент равен нулю. На рисунке 23 представлено сравнение двух функций.



*Рис.23. Сравнение сигмоидальной функции и ReLU.*

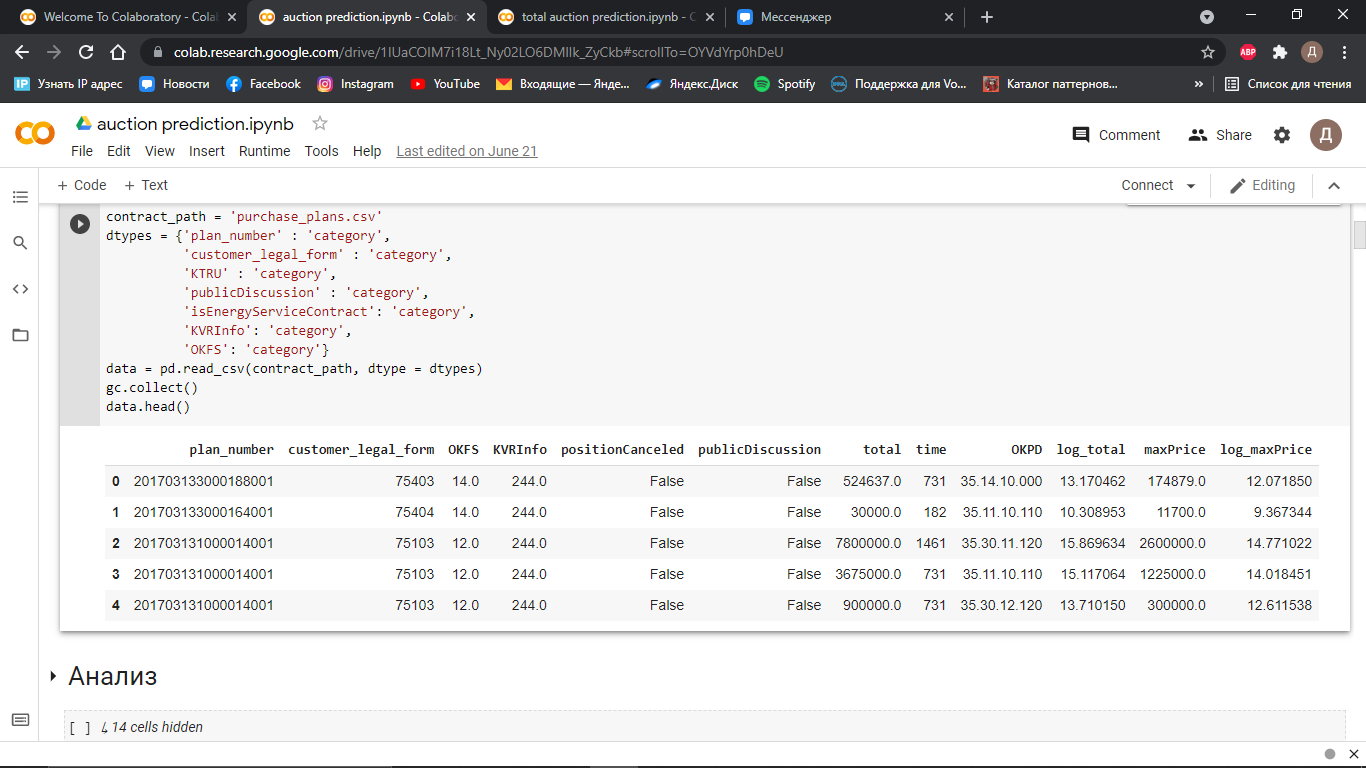
Рассмотрим результат нейронной сети на рисунке 24.



*Рис. 24. Результат нейронной сети.*

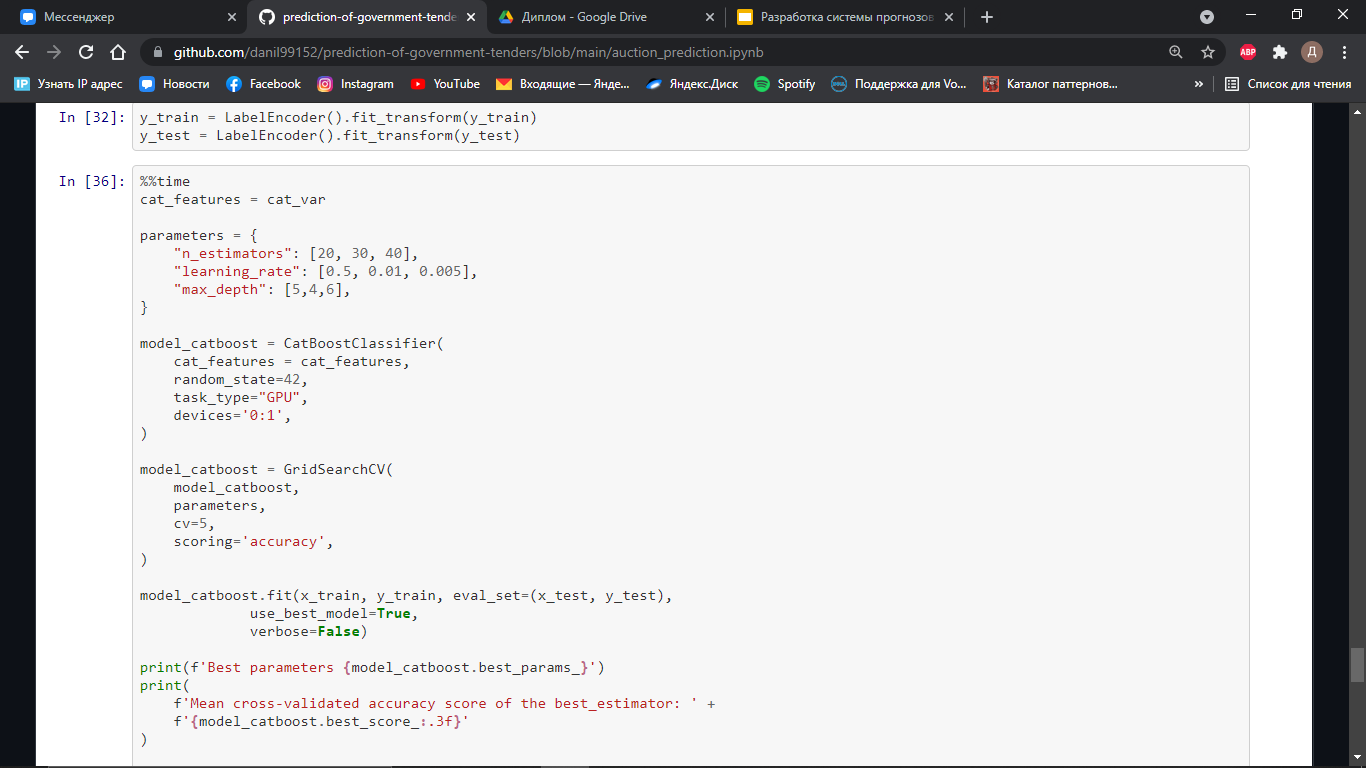
Можно сделать вывод, что нейронная сеть не подходит в качестве классификатора для этой задачи из-за результата кросс-валидации и скорости обучения.

Аналогично для другой задачи задействован алгоритм, а именно для прогнозирования отмены заявки на тендер на этапе аукциона.



*Рис. 25. Набор данных для задач второй и третьей.*

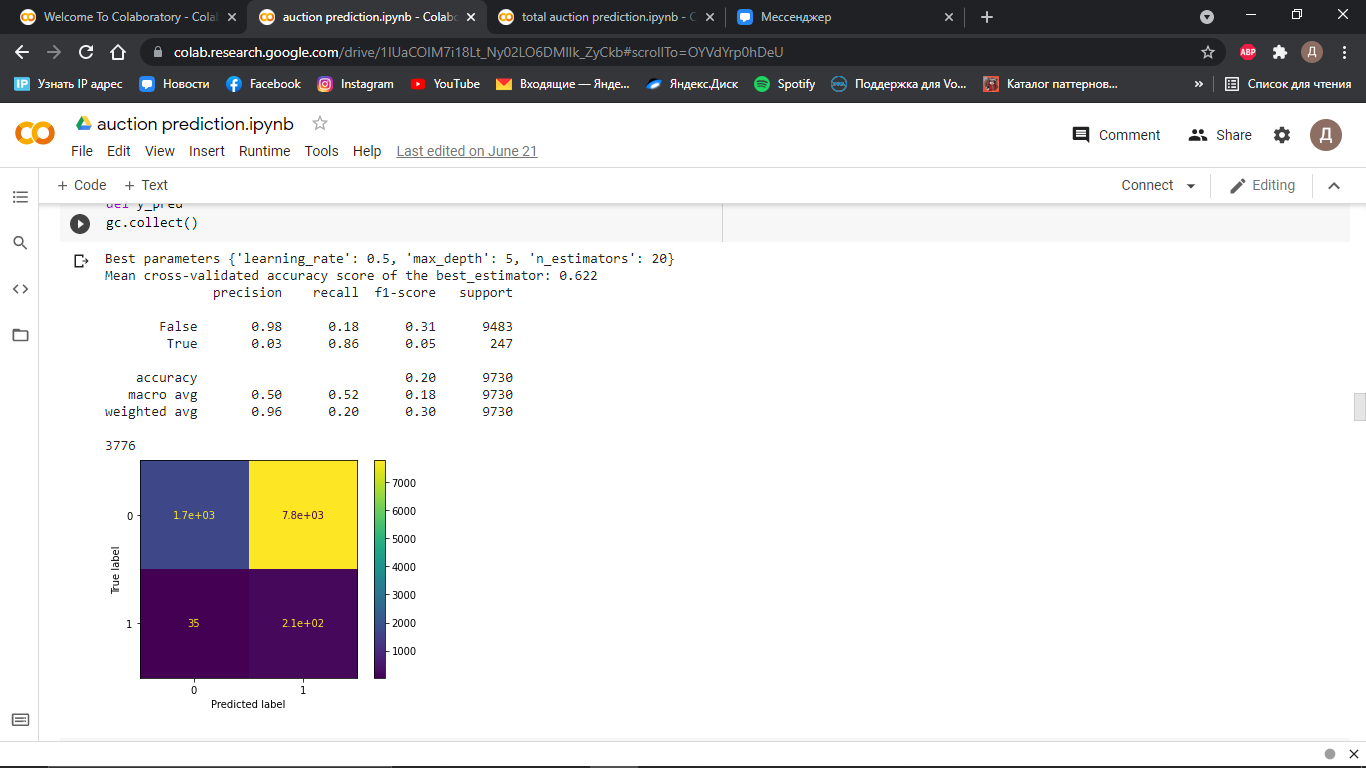
На рисунке 25 представлена таблица из первых 5 объектов из набора данных для следующих двух задач.



*Рис.26. Инициализация CatBoost для предсказания результата аукциона.*

На рисунке 26 можно снова увидеть GridSearchCV и прописанный диапазон параметров для инициализации алгоритма CatBoost.

Снова CatBoost показал лучший результат для предсказания, на рисунке 27 можно это увидеть.

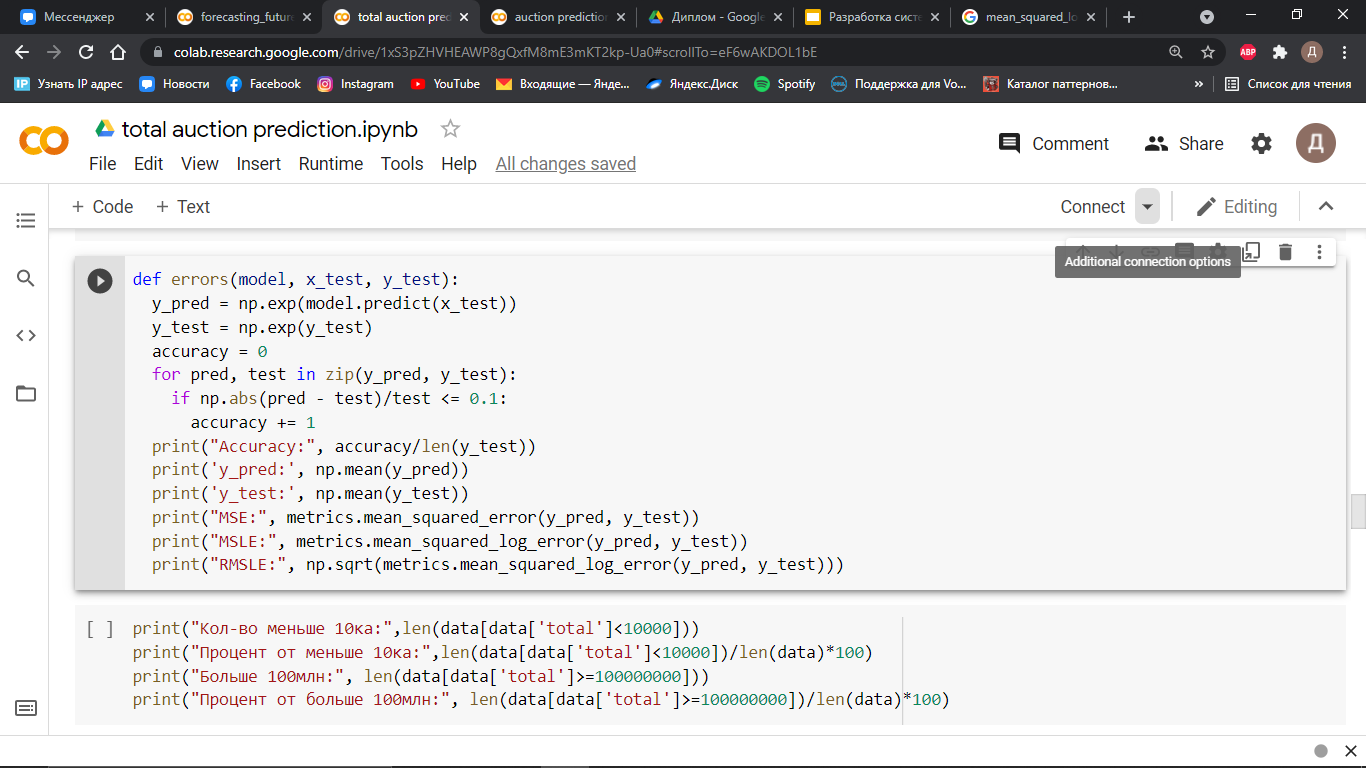


*Рис.27. Подобранные параметры и результат точности CatBoost для новой задачи.*

Аналогично с первой задачей класс риска является класс True, получилось поднять его полноту до 86% без ущерба для точности класса False, для этого пришлось разбить тренировочные данные между классами False и True в отношении 60% при отношении на тестовых данных True к False в 3%.

Третья задача регрессионная. В ней необходимо предсказать цену будущего контракта. Главная проблема задачи в том, что цена контракта разбросана от тысяч до миллиардов рублей. Поэтому было решено взять натуральный логарифм от цены. Натуральный логарифм позволил сократить разброс цены от минимальной до максимальной. Вся обработка данных таким образом позволила избавиться от выбросов и сократить ошибку.

В качестве основной метрики ошибки был взят следующий метод, представленный на рисунке 28.



*Рис.28. Регрессионная метрика ошибки.*

Суть метода заключается в том, чтобы посчитать количество предсказаний, которые отклонялись не больше, чем на 10% от актуальных значений.

Также были использованы метрики средней квадратичной, средней квадратичной логарифмированной и корень средней квадратичной логарифмированной ошибки.

Формула средней квадратичной ошибки:

Формула средней квадратичной логарифмированной ошибки:

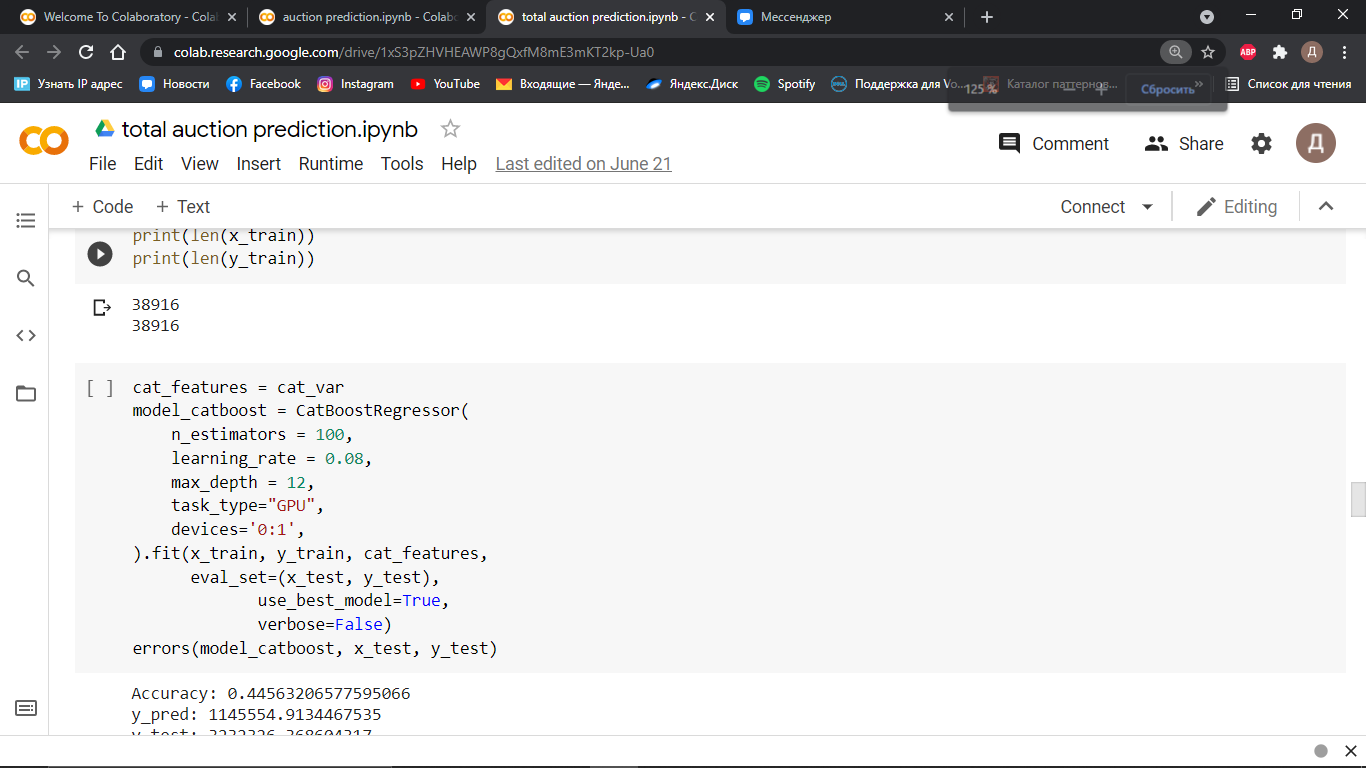
Где - предсказанное значение, – настоящее значение, n – количество значений.

В качестве моделей предсказателей были использованы алгоритмы CatBoost и линейная регрессия. Формула алгоритма линейной регрессии:

.

В формуле можно заметить свободный член (b), угол наклона (a) и переменную, отвечающую за входные данные (x).

Рассмотрим сначала инициализацию регрессионной модели CatBoost на рисунке 29.



*Рис. 29 Регрессионная модель CatBoost.*

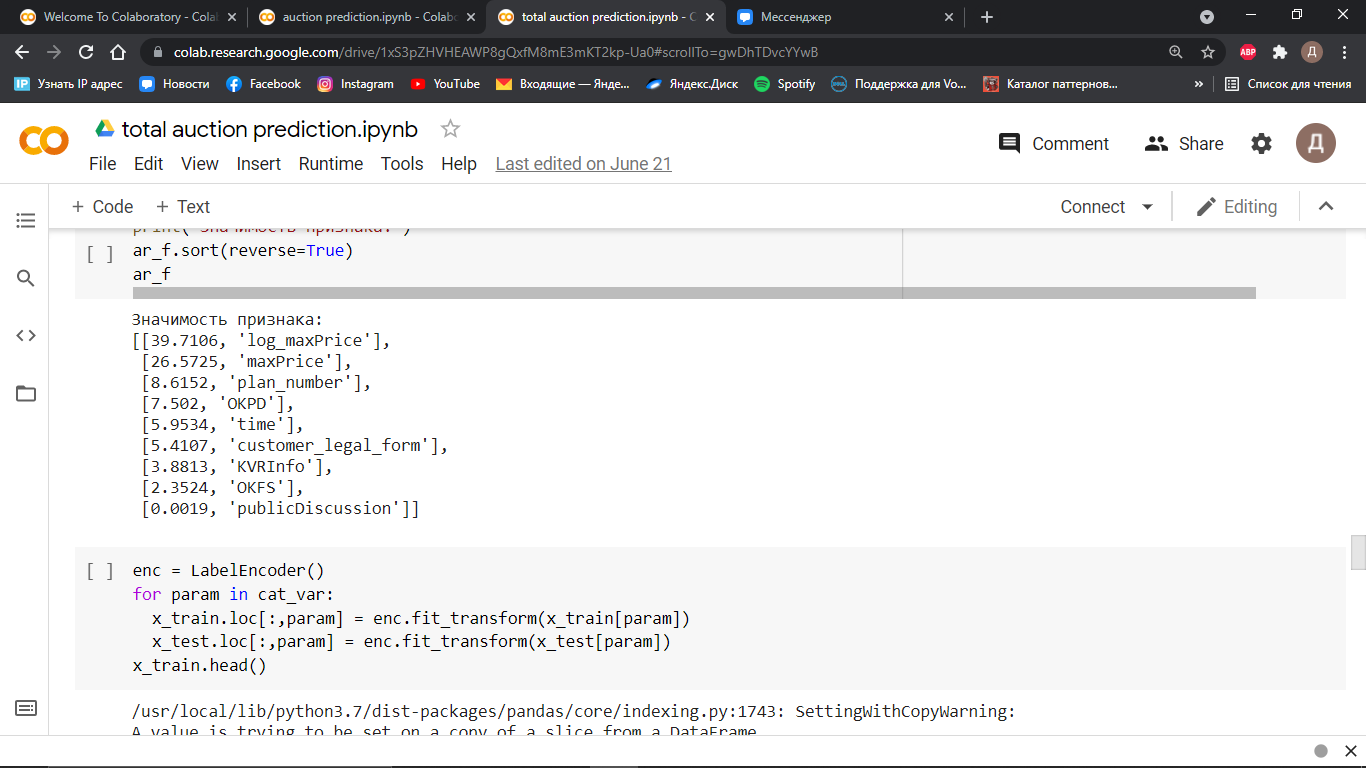
К сожалению, GridSearchCV не работает с регрессионными моделями, поэтому пришлось подбирать параметры модели вручную.

Результат модели следующий:

* Accuracy: 0.44563206577595066
* MSE: 4800267388413611.0
* MSLE: 0.32452282831811935
* RMSLE: 0.569669051571278

Вывод из этого следующий: примерно половина предсказаний удовлетворяет нашему требованию. К сожалению, слишком большой ценовой разброс от тысячных до миллиардных плюс мало данных по каждому контракту, так как каждый контракт индивидуален и требует индивидуальных параметров, а не только типы продукта, тип заказчика и так далее, поэтому предсказание лишь примерное. С ошибкой в 20% всего 63% предсказаний.

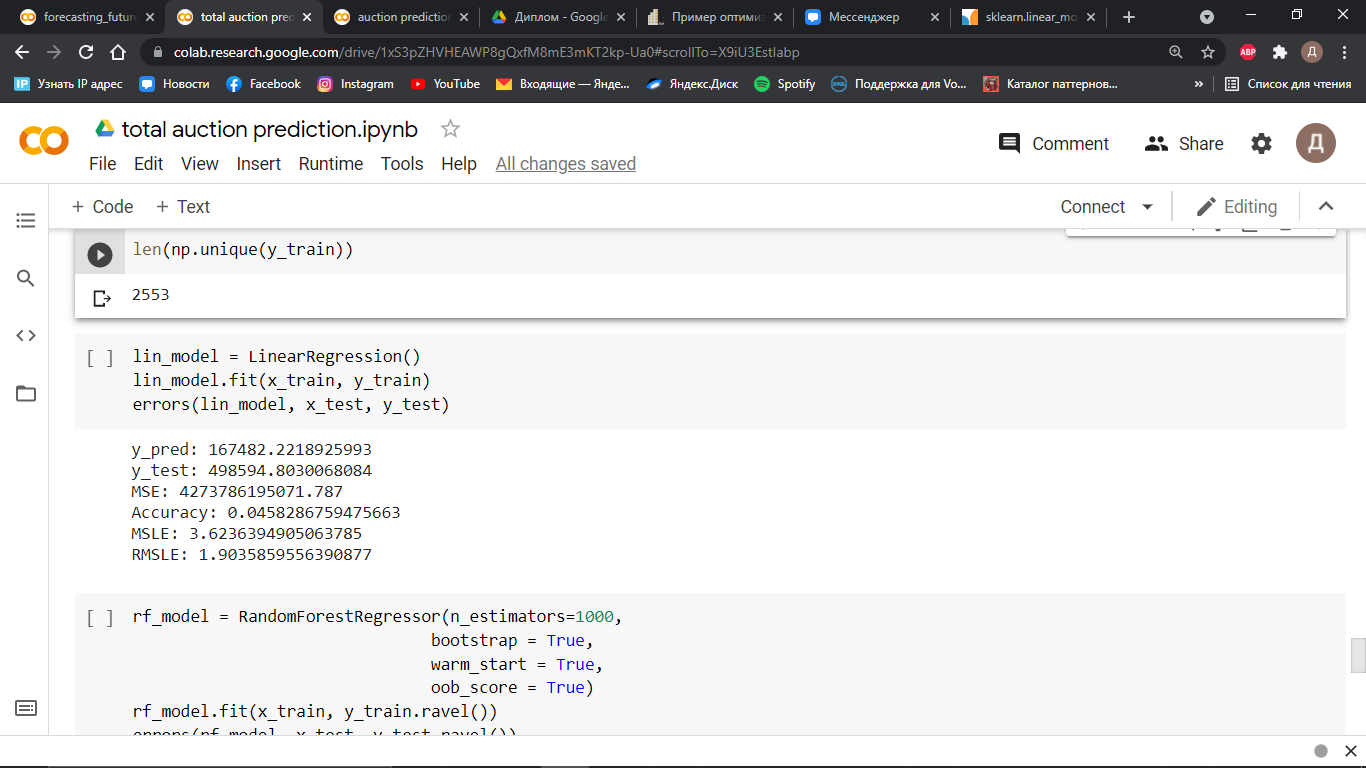
Алгоритмы, построенные на алгоритме дерева решений, имеют возможность дать отчет об важности параметров в предсказании. В CatBoost эту функцию выполняет метод get\_feature\_importance(). Результат такого вывода можно увидеть на рисунке 30.



*Рис. 30. Значимость признаков.*

Рассмотрим следующий алгоритм – линейная регрессия.

Для ее инициализация используется библиотека Scikit-learn. Код для ее инициализации очень простой и представлен на рисунке 31.



*Рис.31. Инициализация, обучение и кросс-валидация линейной регрессии.*

Первая строчка – инициализация, на второй – метод запуска обучения, а на третьей проверка качества предсказания модели.

Результат качества предсказания модели такой:

* Accuracy: 0.22559095580678315
* MSE: 4888029749031195.0
* MSLE: 0.47877614198004026
* RMSLE: 0.6919365158596851

Точность модели на нашей метрике всего 22,6%, что значительно хуже предыдущего алгоритма.

# **Заключение**

В данной работе были созданы модели машинного обучения, предсказывающие риски и цену контракта. Для этого была испробована большая часть популярных современных алгоритмов, решающие эти задачи: от линейной регрессии до нейронной сети и ансамблей деревьев решения, данные были обработаны разными способами, применены различные метрики ошибок.

Естественно, продукт требует доработки как реализованных алгоритмов, так и добавления новых областей прогнозирования. Желательно функционально дорасти до сервисов-аналог, к примеру, добавить предсказание участия компаний-конкурентов в торгах.

# **Литература**

1. 5 видов регрессии и их свойства [Электронный ресурс]: URL: [https://medium.com/nuances-of-programming/5-видов-регрессии-и-их-свойства-f1bb867aebcb](https://medium.com/nuances-of-programming/5-%D0%B2%D0%B8%D0%B4%D0%BE%D0%B2-%D1%80%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D0%B8-%D0%B8-%D0%B8%D1%85-%D1%81%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0-f1bb867aebcb) (дата обращения 15.12.2020).
2. CatBoost против Light GBM против XGBoost [Электронный ресурс]: URL: <https://www.machinelearningmastery.ru/catboost-vs-light-gbm-vs-xgboost-5f93620723db/> (дата обращения 10.06.2021).
3. Гласснер Э. Глубокое обучение без математики. Т. 1: Основы / пер. с анг. В. А. Яроцкого. – М.: ДМК Пресс, 2019.
4. Гласснер Э. Глубокое обучение без математики. Т. 2: Практика / пер. с анг. В. А. Яроцкого. – М.: ДМК Пресс, 2020.
5. Градиентый бустинг – просто о сложном [Электронный ресурс]: URL: <https://neurohive.io/ru/osnovy-data-science/gradientyj-busting/> (дата обращения 25.05.2021).
6. Грас Д. Data Science. Наука о данных с нуля. – СПб.: БХВ-Петербург, 2020.
7. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с анг. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018.
8. Джулли А., Пал С. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow/ пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2018.
9. Метрики в задачах машинного обучения [Электронный ресурс]: URL: <https://habr.com/ru/company/ods/blog/328372/> (дата обращения 23.04.2021).
10. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. – М.: Вильямс, 2017.
11. Николенко С., Кадурин А., Архангельская Е. Глубокое обучение. – СПб.: Питер, 2018.
12. Нишант Ш. Машинное обучение и TensorFlow. – СПб.: Питер, 2019.
13. Орельен Ж. Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем‚ 2-е изд.: Пер. с англ. – СПб.: ООО «Диалектика», 2020.
14. Отличия LabelEncoder и OneHotEncoder в SciKit Learn [Электронный ресурс]: URL: <https://habr.com/ru/post/456294/> (дата обращения 21.02.2021).
15. Паттанаяк С. Глубокое обучение и TensorFlow для профессионалов. Математический подход к построению систем искусственного интеллекта на Python.: Пер. с англ. – СПб.: ООО «Диалектика», 2019.
16. Рашка С., Мирджалили В. Python и машинное обучение. Машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow. – М.: Вильямс, 2019.
17. Случайный лес (Random Forest) [Электронный ресурс]: URL: [https://dyakonov.org/2016/11/14/случайный-лес-random-forest/comment-page-1/](https://dyakonov.org/2016/11/14/%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9-%D0%BB%D0%B5%D1%81-random-forest/comment-page-1/) (дата обращения 14.02.2021).
18. Функции активации нейросети: сигмоида, линейная, ступенчатая, ReLu, tahn [Электронный ресурс]: URL: <https://neurohive.io/ru/osnovy-data-science/activation-functions/> (дата обращения 10.06.2021).
19. Шолле Ф. Глубокое обучение на Python. – СПб.: Питер, 2018.
20. Foreman J. Data Smart. Using Data Science to Transform Information into Insight. Indianapolis: John Wiley & Sons Limited, 2013.
21. The neural network zoo [Электронный ресурс]: URL: <https://www.asimovinstitute.org/neural-network-zoo/> (дата обращения 13.12.2020).
22. Tips Dataset [Электронный ресурс]: URL: <https://www.kaggle.com/sanjanabasu/tips-dataset> (дата обращения 05.02.2021).
23. XGBoost, LightGBM or CatBoost – which boosting algorithm should I use? [Электронный ресурс]: URL: <https://medium.com/riskified-technology/xgboost-lightgbm-or-catboost-which-boosting-algorithm-should-i-use-e7fda7bb36bc> (дата обращения 10.06.2021).