ТЕХНОЛОГИЧЕСКАЯ КАРТА ЗАНЯТИЯ

Тема занятия: Конкурсы на kaggle.com

Аннотация к занятию: обучающиеся изучат платформу Kaggle и примут участие в конкурсе.

Цель занятия: формирование у обучающихся представления о платформе Kaggle, работа с данной платформой и разбор решения задач на платформе и вне её.

Задачи занятия:

- изучить платформу Kaggle;
- узнать, что такое соревнования по машинному обучению;
- познакомить с датасетом и загрузкой решений на платформу Kaggle;
- решить задачи на платформе, поучаствовать в конкурсе.

Ход занятия

Этап занятия	Время	Деятельность педагога	Комментарии, рекомендации для педагогов
Организационный этап	5 мин.	Добрый день! Мы познакомимся с Kaggle — платформой для конкурсов по машинному обучению. Поговорим о том, как устроены конкурсы, пройдёмся по сайту Kaggle, изучим его интерфейс и возможности. Затем построим пайплайн решения задачи «Титаник». Пройдём путь от скачивания датасета со страницы конкурса до получения ответов модели на тестовой выборке. Загрузим решение на платформу и узнаем своё место в рейтинге. После этого вы сможете самостоятельно участвовать в любых конкурсах по машинному обучению	Приветствие. Создание в классе атмосферы психологического комфорта
Постановка цели и задач занятия. Мотивация учебной деятельности обучающихся	7 мин.	Тема занятия — «Конкурсы на Kaggle.com». Как вы думаете, какие задачи мы сможем сегодня решить? **Возможные ответы обучающихся:** ● поработаем на платформе Kaggle, ● решим задачи, ● примем участие в конкурсе. Мы научимся работать с Kaggle, поэтому обработке датасета и выбору параметров мы уделим не так много времени.	Способствовать обсуждению мотивационных вопросов

		После просмотра вы сможете улучшить пайплайн самостоятельно и добиться лучших результатов в конкурсе.	
Изучение нового материала	50 мин.	Начнём. Заходим на сайт kaggle.com. Чтобы участвовать в соревнованиях и пользоваться другими преимуществами платформы, нужно зарегистрироваться. Процесс регистрации очень прост. **Вопрос для обсуждения** Как вы думаете, что такое соревнования по машинному обучению? **Ответы обучающихся** Соревнование — это когда перед вами ставится задача машинного обучения, и к ней даются тренировочный и тестовый датасеты. Задачей может быть предсказание цены дома по его характеристикам. Ответы к тестовому датасету при этом не даны. Ваша задача — обучить модель машинного обучения на тренировочной части и получить предсказания на тестовой части. Эти предсказания вы отправляете на сайт соревнования, где они сравниваются с правильными ответами по некоторой метрике, например, MSE, и вам выдаётся ваш результат. Вместе с вами в соревновании участвуют и другие люди. Ваша задача — обучить такую модель, чтобы она выдавала лучшие значения метрики качества на тестовой выборке, чем у ваших конкурентов. За лучшие решения во многих соревнованиях дают призы. Соревнования могут идти от нескольких недель до нескольких лет, и за время соревнования вы можете	

		проводить много экспериментов с вашей моделью и много раз отправлять решения в систему. Организуют такие соревнования чаще всего компании. Например, Google, Microsoft и любые другие. Такие соревнования — это хороший пиар и возможность найти новые идеи решения задачи, которую они дают в соревновании. Kaggle — одна из платформ, на которой проводятся такие соревнования. Кроме неё существует много других платформ, но Kaggle — это, наверное, сама крупная и известная из них. Конкурсы на ней устроены довольно удобно, и, помимо участия в соревнованиях, Kaggle предоставляет пользователям дополнительные возможности. Давайте детально посмотрим на то, как устроены соревнования по машинному обучению на платформе Kaggle. Доступные соревнования находятся во вкладке Competitions слева в меню. Нажмём. Видим список доступных в данный момент соревнований. Рядом с каждым указано, сколько ещё времени будет идти соревнование и какой у него призовой фонд. К текущим соревнованиям можно присоединяться в любой момент, даже если до его конца осталась пара дней. Хотя, конечно, это не всегда стоит того: за пару дней вряд ли получится обучить хорошую модель для победы. Поэтому стоит выбирать те соревнования, в которых осталось достаточно времени для участия.	

		На некоторых соревнованиях вместо оставшегося времени указано ongoing. Это значит, что соревнование открыто бесконечно: у него нет времени завершения или организаторы еще не решили, когда его завершить. Такие соревнования без конца — это обычно обучающие конкурсы. Они созданы специально для людей, которые только начинают изучение машинного обучения и хотят попробовать свои силы в простом соревновании. А ещё это отличная возможность для новичков познакомиться с платформой Kaggle и научиться с ней работать. Как раз для нас. Давайте откроем одно такое обучающее соревнование. Это «Титаник». В нём требуется решить задачу классификации: по информации о пассажирах лайнера «Титаник» определить, выжил пассажир после крушения или нет. На экране вы видите, как выглядит главная страница конкурса. Страницы всех соревнований устроены одинаково. Давайте пройдёмся по этому соревнованию и посмотрим, что тут есть. На главной странице обычно размещают общие слова о том, что это за конкурс. Например, здесь написано, что «Титаник» — это обучающий конкурс для тех, кто хочет погрузиться в мир машинного обучения. Ниже идёт немного информации о том, что такое лайнер «Титаник» и что в этом конкурсе вам нужно научиться предсказывать, кто из пассажиров выжил, а кто — нет. Наверху мы видим организатора конкурса, количество команд-участников и сколько времени осталось до конца.	

		Это обучающее соревнование, поэтому времени конца нет, указано ongoing. Организатор этого соревнования — сам Kaggle. Перейдём во вкладку data. В этой вкладке обычно находится описание данных, которые даются в соревновании, тут же эти данные можно скачать. В этом соревновании данные представлены в знакомом нам виде: файлы train.csv и test.csv. Можно даже посмотреть на данные прямо здесь, не скачивая их. Нажмём на train.csv — справа покажется часть таблицы. Видим здесь признаки, колонка survived — это целевая переменная. Для каждого человека указано, выжил он при крушении лайнера или нет: 0, если нет, 1, если выжил. Кроме train и test тут есть ещё один файл — gender_submission. Зачем он нужен, мы узнаем позже, когда будем учиться отправлять ответы нашей модели в это соревнование. Развернём описание данных. Как правило, здесь описывается, что представляют собой данные и как с ними работать. Здесь как раз написано, что данные разбиты на две части: train и test, и что участник соревнования должен обучать свою модель на тренировочной части данных, и отправить на конкурс предсказания модели на тестовой части. Ниже дана таблица с описанием всех колонок в данных: название каждой колонки и то, что эта колонка означает. Мы ещё вернемся сюда, когда в следующем видео скачаем эти данные и будем обучать на них модель.	

		Чтобы скачать данные, нужно нажать на кнопку download all. Вернёмся на секунду во вкладку overview. Здесь в меню слева есть вкладка evaluation. В ней указано, с помощью какой метрики качества будут оцениваться ваши решения. Например, для задачи регрессии это может быть метрика MAE или MSE, для задачи классификации — accuracy, F1, ROC AUC и другие. Здесь написано, что в этом соревновании метрика качества — accuracy. Ниже также показано, в каком виде нужно отправлять решения на платформу. Решение — это ответы вашей модели на тестовой выборке. Тут видно, что решение должно быть в виде файла, в котором есть два столбца с названиями PassengerId и Survived. Далее каждая строка — это ответ вашей модели на одного человека из тестовой выборки. В первой колонке — его id, он берётся из тестовых данных. Во второй колонке — ответ модели для человека: 0, если ваша модель считает, что человек не выжил, и 1, если выжил. Как формировать ответы модели в таком виде и отправлять их на Kaggle, мы узнаем немного позже. Итак, мы разобрались с тем, где брать данные для обучения модели и данные о том, как решения будут оцениваться. Теперь давайте перейдём во вкладку leaderboard. Это таблица участников соревнования, отсортированная по метрике качества решения. Сверху находятся имена людей, которые отправили лучшие решения по метрике качества. Когда вы отправляете свои ответы на тестовой выборке на платформу, внутри по вашим ответам и правильным ответам	

		считается метрика качества. Вам выдаётся результат. Вы можете видеть, на каком месте таблицы лидеров вы находитесь. Эта колонка — значение метрики качества решения, следующая колонка — количество отправок решений этим участником, последняя колонка — сколько времени назад этот человек отправил своё последнее решение. Если вы отправляете несколько решений в соревнование, то в таблице лидеров показывается лучший результат. В этом соревновании видно, что лидеры получили метрику accuracy, равную 1. Это значит, что их предсказания на тестовой выборке были идеальны. Так как это обучающее соревнование, которое длится бесконечно, в этом нет ничего удивительного: датасет «Титаник» исследован вдоль и поперёк. Люди уже давно смогли подобрать идеальную модель для решения этой задачи. В реальном соревновании вы, кончено, вряд ли увидите идеальное значение метрики в таблице лидеров. В соседней вкладке у меня открыто другое соревнование на Kaggle от компании Google. Это реальное соревнование по машинному обучению. Видно, что в нём сейчас 643 команды и что конкурс будет идти ещё 2 месяца. Если посмотреть во вкладку leaderboard, то видно, что результаты участников далеки от единицы. Хотя метрика качества, как мы видим, тоже accuracy. Пока мы находимся на странице реального соревнования от Google, отметим ещё вот что. Во вкладке leaderboard есть две кнопки — public и private. Что это такое?	

		Смотрите, в реальных соревнованиях, когда вы отправляете ответы вашей модели на тестовой выборке в систему, метрика качества для вашего решения считается только от половины элементов датасета. Пусть в тестовом датасете 1000 элементов. Вы получили ответы для этой 1000 элементов, записали в файл и отправили в систему. Система возьмёт 500 ваших ответов и 500 соответствующих им правильных ответов и посчитает от них метрику качества. И значение метрики качества на этих пятистах элементах вы и увидите в таблице во вкладке public. То, насколько хорошо ваша модель предсказала ответы для остальных пятисот элементов выборки, вы узнаете только после окончания соревнования. Значение метрики качества для них будет доступно во вкладке private. Мы видим таблицу лидеров и значения их метрик качества во вкладке public: то есть для половины тестового датасета. Если мы заглянем во вкладку private, здесь будет пусто. Метрики здесь появятся только после окончания конкурса через два месяца. Зачем же так делается? А затем, чтобы участники не смогли схитрить. Смотрите: задача участника конкурса — честно обучить свою модель только на обучающей части данных, получить предсказания на тестовых данных и отправить в систему. Допустим теперь, что участник знает, какую метрику его модель получила на всём тестовом датасете. Тогда он может просто обучить много разных моделей с разными параметрами под задачу, получить от них ответы на тестовую выборку и отправить их в систему. А потом в качестве своего финального решения выбрать ту, которая показала наилучший результат. И тогда получится, что	

		участник поступил нечестно: не думал и подбирал параметры своей модели на основе идей, а просто случайно получил модель, у которой вышел хороший результат. Возможно даже, что эта модель не так уж и хороша на самом деле. Возможно, просто так случайно вышло, что именно на этих тестовых данных она получила хороший результат. А на других данных получила бы результат намного хуже. Не хочется, чтобы в соревновании побеждал человек, модель которого случайно получила лучший скор. Поэтому тестовая выборка делится на public и private часть. Участникам во время соревнования показывается качество их решения только на public, публичной части. В этом случае даже если участник подобрал такую модель, что она даёт хороший скор на публичной части, это не гарантирует, что метрика на приватной части будет так же хороша. Это заставляет участников обучать модели, которые были бы стабильны: в которых участники были бы уверены, что они покажут хороший результат на всём тестовом датасете, а не только на его public части. Как правило, это значит, что такие модели в целом хорошо решают поставленную задачу. Можно смотреть на это ещё с такой точки зрения: public часть — это тестовый датасет, а private часть — это реальный мир. Когда вы обучаете модель машинного обучения и тестируете её на тестовых данных, ваша цель — не чтобы ваша модель просто хорошо предсказывала тестовые данные, а чтобы она в принципе хорошо работала. Чтобы её можно было использовать и далее на других, новых данных. Private часть — это имитация таких «новых данных». Если ваша модель хорошо работает и на public, и на private, значит, она действительно хорошо справляется с	

		поставленной задачей, не переобучилась на тестовых данных, и за неё можно дать вам приз в конкурсе. Вот что важно: после окончания соревнования ваше итоговое место в таблице лидеров определяется только по приватной части. Метрика считается только по второй половине датасета. Из-за этого легко может быть такое, что положение участников таблицы после окончания соревнования сильно меняется: некоторые участники могут сильно упасть, другие — сильно подняться. По той же причине в конкурсах ограничено число решений, которые участники могут отправить в день, — от двух до пяти. Если разрешить отправку неограниченного количества предсказаний в день, можно будет просто перебрать все возможные модели и параметры к ним и найти те, благодаря которым модель будет выдавать идеальные ответы на тестовую выборку. Опять же, это совершенно не будет значить, что модель в принципе хорошо работает: мы просто случайно нашли то, что выдаёт хороший результат именно для этой выборки. Если зайти во вкладку с правилами, видно, что в день можно отправлять только 5 ответов. Вернёмся теперь к нашему конкурсу «Титаник». Здесь деления на public и private части в таблице нет, потому что конкурс учебный и длится бесконечно. Тут вы сразу будете видеть метрику качества вашего решения на всём тестовом датасете.	

		Идём дальше. Вкладка Rules. Здесь описываются правила соревнования. Например, чем можно пользоваться, а чем нельзя, сколько максимум посылок в день можно совершать, когда соревнование заканчивается и подобное. Также здесь обычно написано, можно ли в конкурсе участвовать в команде и какое максимальное число человек в команде может быть. Участие в команде — это когда вы ещё с несколькими людьми объединяетесь и начинаете решать задачу вместе. В таблице лидеров вы будете стоять в одной строке как команда. Чтобы пригласить кого-то в свою команду или принять приглашение друга в его команду, нужно зайти во вкладку team. У меня в этом конкурсе нет команды, поэтому здесь показана только я одна. Остались две вкладки: code и discussion. Discussion — это форум, где люди могут общаться на тему конкурса. Задавать вопросы, отвечать на них и подобное. Если у вас есть проблема или вопрос по конкурсу, можно зайти сюда, может быть, похожий вопрос уже кто-то задал. Если нет, можно задать самому. Code — это вкладка, где люди делятся примерами своего кода на тему этого соревнования. В некоторых конкурсах делиться кодом нельзя, но обычно всегда можно. В конце концов, если кто-то поделился своим кодом хорошего решения, плохо от этого может стать только ему самому, если кто-то возьмёт его решение и обгонит в соревновании. В «Титанике» масса кода, так как это учебное соревнование. Нажмём, например, на первый код. Видим, что он выглядит как Jupyter Notebook: всё очень красиво и понятно оформлено.	

		Вкладка code разных соревнований — это отличный инструмент для обучения. Можно смотреть, как другие люди решают ту или иную задачу, получать новые идеи и опыт, пытаясь это повторить или развить идею дальше. Все страницы конкурсов на Kaggle устроены одинаково: одни и те же вкладки и информация в одинаковых местах. Как отправить решение задачи на Kaggle, мы узнаем позже. Другие платформы для соревнований по машинному обучению выглядят по-другому, но функционал у всех при этом примерно такой же. У них есть вкладки с данными, таблица лидеров, разделённая на public и private, форум для обсуждений и подобное. Поэтому, разобравшись с Kaggle, вы легко освоите и все остальные площадки. На Kaggle есть ещё две полезные функции кроме самих соревнований. Первая — это датасеты. Зайдём во вкладку datasets. Это место, где собраны тысячи разных датасетов для разных задач, и они постоянно обновляются и пополняются. Например, хотите датасет для классификации картинок яблок — я уверена, он тут найдется. Так что если вам когда-то будет нужно найти датасет для своего проекта или тренировки по машинному обучению, то Kaggle — лучшее для этого место. Здесь можно искать датасеты по ключевым словам и фильтрам.	

		Второй функционал Kaggle — это ноутбуки. Kaggle — это почти Google Colab. Здесь тоже можно бесплатно писать код в ноутбуках и запускать его. Чтобы создать ноутбук, нажмём create, new notebook. Как видим, интерфейс похож на Colab, разобраться несложно. Из плюсов ноутбуков на Kaggle — их удобно переносить во вкладку code соревнований. Сейчас мы поработаем над задачей предсказания и попробуем угадать, выжил ли пассажир «Титаника» или нет, по его характеристикам. **План такой:** ● скачать и загрузить данные в Colab; ● предобработать тренировочную часть и обучить на ней модель машинного обучения; ● предобработать тестовую часть и получить предсказания модели на ней; ● записать эти предсказания в файл, как требует этого Kaggle, и отправить их в конкурс на сайт. Начнём. Переходим на сайт https://www.kaggle.com/, идём во вкладку Data и нажимаем «Скачать всё» (download all). Скачается zip-архив с данными. Идём в файловый менеджер и распаковываем архив. Получим папку с тремя файлами: теми, что мы видели на странице конкурса: train.csv, test.csv и gender_submission.csv. Вернемся а Colab и загрузим все три файла. Отлично. Теперь можно кодить.	

		Как обычно, импортируем нужные библиотеки и считаем обучающие данные с помощью pandas read csv. Чтобы получить предсказания модели, которые мы сможем послать на Kaggle, нам нужно предобработать датасет и обучить модель. После занятия вы сможете сами улучшить в код предобработки данных и выбора модели, чтобы получить лучшую метрику качества на Kaggle. Давайте начнём. Разделим выборку на признаки и целевую переменную. Целевая переменная — survived, она показывает, выжил пассажир или нет. В data у нас будут признаки, в y — целевая переменная. Выведем информацию о наших данных: data.info(). Здесь видим, во-первых, что у нас есть колонки с пропусками. Это age, cabin и embarked. Также видим, что у нас есть категориальные признаки, которые представлены не числами, а строками. Чтобы обучить модель на этих данных, нужно обязательно избавиться от пропусков и перевести категориальные переменные в числовые. Этим сейчас и займемся. Во-первых, давайте удалим ненужные признаки: 'PassengerId', 'Name', 'Ticket', 'Cabin'. Это признаки, которые, кажется, не несут никакой полезной информации для модели. Не помогут ей никак определить, выжил человек на «Титанике» или нет. Например, passenger id — это уникальный код пассажира, он даётся человеку абсолютно случайно. Name — имя пассажира, не думаю, что от имени зависело, выживет человек или нет. Тикет — номер билета	

		пассажира. Если мы посмотрим в данные, увидим, что номер билета тоже уникален для всех пассажиров, как и айди. Возможно, конечно, из номера билета мы что-то можем понять полезное о пассажире: например, сколько билет стоит и в каком классе каюты человек ехал, но, во-первых, у нас в датасете есть отдельные признаки «класс» и «стоимость билета», а во-вторых, сейчас мы ничем сложным заниматься не будем. То же самое и с переменной cabin: это номер каюты пассажира. С первого взгляда он не несёт много полезной информации, а еще в нём, как мы видели выше, есть пропуски. Поэтому просто его удалим. Запустим ячейку. Посмотрим, какие пропуски и категориальные переменные у нас остались после этого. Видим: пропуск в age и embarked и категориальные переменные sex и embarked. Колонка age — это возраст пассажира. Заполним в ней пропуски средним значением возраста пассажиров в обучающих данных. Опять же: это, наверное, не лучший способ заполнения пропусков, но мы сейчас за идеалом не гонимся. Колонка Embarked — это порт, а котором пассажир сел на «Титаник». «Титаник» забирал людей из трёх портов. Давайте посмотрим, сколько людей из тренировочного датасета сели на «Титаник» в каждом из портов. Выведем value counts. Видим, что самый популярный порт — S. S — это southampton. Давайте заполним пропуски в колонке значением S.	

		Посмотрим, что у нас вышло. Убедимся, что пропусков в данных больше не осталось. Да, пропусков больше нет. Осталось привести категориальные переменные в числовой вид. У нас есть два категориальных признака: sex и embarked. Давайте переведем их в числовые с помощью LabelEncoder из Sklearn: Заводим первый labelencoder для переменной пол (sex). Делаем fit и transform. Получаем новый столбец sex, в котором вместо букв числа. То же самое с embarked. Заводим для него отдельный labelencoder и переводим этот столбец в числа. Снова посмотрим на данные и убедимся, что категориальных переменных не осталось. Да, вс` хорошо. Можно обучать модель. Давайте обучим логистическую регрессию и KNN на нашем датасете. Так как у нас нет ответов для тестовой выборки, оценим качество этих двух моделей с помощью кросс-валидации. Потом, когда отправим наши ответы для тестового датасета на Kaggle, сравним метрику, полученную для тестовой выборки на Kaggle, и то, что мы получим сейчас на кросс-валидации.	

		Подбирать параметры моделей мы сейчас не будем. Вы можете этим заняться дома и постараться улучшить пайплайн. Начнеё с логрегрессии. Заводим её и с помощью cross val score посчитаем кросс-валидацию на данных. Передаём в cross val score нашу модель lr, данные data и y, ставим количество фолдов равное пяти и метрику, по которой оценивать качество, — accuracy. Запускаем. Вот что получилось. Проделаем то же самое с KNN. Хорошо, мы получили оценки того, какой должна быть метрика качества accuracy на тестовых данных у двух этих моделей. Обучим теперь эти модели на наших тренировочных данных: lr.fit(), knn.fit(). Отлично, модели обучили. Осталось предобработать тестовые данные и получить на них ответы моделей. Посмотрим на наши тестовые данные: загрузим их. Видим, что в них есть всё те же колонки, что в тренировочных данных, кроме колонки survived — целевой переменной. Её значения нам нужно предсказать.	

		Предобрабатывать тестовые данные нужно ровно так же, как мы это делали с тренировочными. Нужно удалить те же колонки, заполнить пропуски в столбцах теми же значениями, точно таким же образом перевести категориальные признаки в числовые. Во-первых, удалим те же 4 колонки. Во-вторых, заполним пропуски в колонках age и embarked точно теми же значениями, которыми заполняли пропуски в обучающей части. Fill_value здесь — это среднее значение колонки age в тренировочных данных. Проверим, что пропусков в тестовых данных не осталось. Смотрите, что тут: в колонке fare есть пропуск. Такое бывает: в тренировочных данных в колонке fare пропуска не было, а в тестовых он есть. Его тоже нужно заполнить, иначе модель не выдаст нам ответ. Заполним пропуск в колонке Fare средним значением колонки из тренировочной части выборки. Так, теперь пропусков больше нет. Переходим к переводу категориальных признаков в числовые. Берём те же самые обученные labelencoders для колонок sex и embarked и трансформируем ими значения тестовых колонок в числа. Это всё, что мы сделали с обучающими данными. Давайте выведем информацию о тестовых данных, чтобы убедиться, что всё готово.	

		Да, всё хорошо: пропусков нет, категориальных переменных тоже. Можем получать предсказания моделей. Получим предсказания обеих моделей: LR и KNN. Посмотрим, как выглядят предсказания: как и ожидалось, это массив из нулей и единиц. Теперь нам нужно получить файл с ответами для отправки на конкурс. Тут-то нам и пригодится файл gender submission csv, который мы скачали вместе с train.csv и test.csv. Давайте его откроем. Структура этого файла такая же, как та, что мы видели в разделе evaluation, когда изучали конкурс «Титаник». То есть этот файл — шаблон того, как нужно отправлять решения на Kaggle. Нам нужно просто заменить колонку Survived на вектор ответов нашей модели, сохранить этот файл и отправить его на конкурс. Давайте это сделаем. Сохраняем файл как submission.csv. Он отобразится у нас тут. Теперь мы можем скачать его на компьютер. Скачали. Переходим на сайт на страницу нашего конкурса. Здесь есть кнопка submit prediction. Жмём. Попадаем на страницу загрузки файла с ответом. Давайте загрузим. Отлично. Ниже есть поле для комментария. Тут вы можете написать, что за файл вы отправили. Например, что это была за модель, которая выдала такие предсказания, какие у неё были параметры. Эти комментарии будете видеть только вы. Зачем комментарии нужны: чаще всего после нескольких недель или месяцев конкурса у вас будет отправлена куча	

		разных файлов с предсказаниями разных моделей. Все они будут отображаться в одном месте. Комментарии нужны, чтобы ориентироваться в этих файлах. Давайте напишем в этом поле, что у нас модель lr baseline. Baseline будет значить, что это самая простая модель, у которой мы не подбирали гиперпараметры и не предобрабатывали данные. Отправим решение в систему. Итак, мы прошли путь от знакомства с датасетом до загрузки решения на Kaggle. Пайплайн, который мы разобрали, подходит и для задач вне платформы — у них такая же логика. Это значит, что вы можете участвовать в любых конкурсах по машинному обучению. Также мы подготовили обучающий конкурс на Kaggle.	
Закрепление изученного материала	15 мин.	**Вопросы для обсуждения** ● Что собой представляет платформа Kaggle? ● В чём суть соревнований по машинному обучению? Какие конкурсы существуют? ● Как загрузить решение задачи на платформу Kaggle?	Педагог организует беседу по вопросам
Этап подведения итогов занятия (рефлексия)	8 мин.	**Вопросы для обсуждения** ● Чему я научился? ● С какими трудностями я столкнулся?	Педагог способствует размышлению обучающихся над вопросами

		• Каких знаний мне не хватает для более глубокого понимания изученного материала? • Достиг ли я поставленных целей и задач?	
Информация о домашнем задании, инструктаж по его применению	5 мин.	Это домашнее задание будет посвящено полноценному решению задачи машинного обучения. **Есть две части этого домашнего задания:** • Сделать полноценный отчет о вашей работе: как вы обработали данные, какие модели попробовали и какие результаты получились (максимум 10 баллов). За каждую выполненную часть будет начислено определенное количество баллов. • Лучшее решение отправить в соревнование на kaggle (максимум 5 баллов). За прохождение определенного порогов будут начисляться баллы. Обе части будут проверяться в формате peer-review. Т.е. вашу посылку на степик будут проверять несколько других студентов и аггрегация их оценок будет выставлена. В то же время вам тоже нужно будет проверить несколько других учеников. Пожалуйста, делайте свою работу чистой и понятной, чтобы облегчить проверку. Если у вас будут проблемы с решением или хочется совета, то пишите в наш чат в телеграме. Во всех пунктах указания это минимальный набор вещей, которые стоит сделать. Если вы можете сделать какой-то шаг лучше или добавить что-то свое --- дерзайте!	

		Как проверять? Ставьте полный балл, если выполнены все рекомендации или сделано что-то более интересное и сложное. За каждый отсутствующий пункт из рекомендация снижайте 1 балл. Метрика Перед решением любой задачи важно понимать, как будет оцениваться ваше решение. В данном случае мы используем стандартную для задачи классификации метрику ROC-AUC. Ее можно вычислить используя только предсказанные вероятности и истинные классы без конкретного порога классификации + она работает даже если классы в данных сильно несбалансированны (примеров одного класса в десятки раз больше примеров другого). Именно поэтому она очень удобна для соревнований.	

Рекомендуемые ресурсы для дополнительного изучения:

1. Платформа Kaggle. [Электронный ресурс] – Режим доступа: https://www.kaggle.com/.
2. Знакомство с Kaggle: изучаем науку о данных на практике. [Электронный ресурс] – Режим доступа: https://tproger.ru/translations/kaggle-competitions-introduction/.
3. Лучшие в Kaggle: что такое соревновательный дата-сайенс и как достичь в нем успеха. [Электронный ресурс] – Режим доступа: https://habr.com/ru/company/skillfactory/blog/529308/.
4. Что такое Kaggle соревнования. [Электронный ресурс] – Режим доступа: https://career.i-neti.ru/chto-takoe-kaggle/.