**Добрый день, Степан! Практически по всем пунктам критериев Вы идеально выполнили задание, замечательная и красивая работа! :) Требования к оформлению: + выполнение в Jupyter Notebook в соответствии с ноутбуком-шаблоном; + структура оформления (отформатированные выводы в отдельных ячейках типа MarkDown, хорошо оформленный лаконичный код, ячейки сделали наглядными, удобно и понятно разбирать Ваше решение); + широкое использование пройденных в курсе библиотек, ничего лишнего; + читаемый и понятный код, отдельно хочется отметить грамотно продуманные имена переменных и функций; + оформили графики по всем правилам, плюс за содержательные названия и подписи осей. По заданиям: 0). Здорово, что построили несколько графиков для разведывательного анализа, распределения отдельных признаков зачастую о многом говорят. Чтобы упросить процесс и сразу получить все возможные попарные связи признаков, а также их распределения – можно использовать функцию pairplot из библиотеки seaborn (особенно важно исследовать распределение таргета). Про мультиколлинеарность признаков можно узнать из карты корреляций. 1). Верно рассчитали новые признаки и закодировали категориальные фичи. В принципе, признак пола здесь не слишком многочисленный по количеству различных значений, но тем не менее при таком типе кодирования может возникнуть проблема появления некоторого отношения порядка между признаками – 0 < 1 === Male < Female. Поэтому, на практике чаще используют OneHotEncoder (или get\_dummies из библиотеки pandas, чтобы не возникало необходимости преобразовывать данные из numpy-массива обратно в pandas). 2). Правильно, что выбрали RobustScaler, в данных, действительно, есть выбросы, которые мы не очищаем – следовательно, нужно воспользоваться именно им. Здорово, что не ошиблись и обучили скейлер только на тренировочных данных, а на тестовых только трасформировали. Отдельный плюс за проведённое мини-исследование перед выбором способа масштабирования. 3). Совершенно верно выбрали метрику, отличное содержательное обоснование, нечего добавить☺ 4). Здорово, что сравнили метрики модели на стандартизованных и обычных данных, наглядно можно проследить, насколько важен процесс масштабирования☺ Отдельный плюс за тестирование различных гиперпараметров. Модель, действительно, ближе к недообученной, поэтому регуляризация здесь не повлияет на качество. Можно также построить графики зависимости метрики от различных значений С. 5). Аналогично хорошо, что попробовали разные гиперпараметры и построили график. Несмотря на расхождение метрик на тренировочных и тестовых данных всё-таки они выше, чем у обычной логистической регрессии – можно сделать вывод, что мы движемся в правильном направлении, решая проблему недообучения. 6). Тоже всё правильно. Интересно наблюдать, как с каждым улучшением метрика увеличивается☺ 7). К обучению модели вопросов нет, отлично, что отметили переобучение – действительно, деревья очень склонны к этому, особенно, если делать их максимально глубокими и не добавлять методы регуляризации. 8). Здорово, что получилось увеличить метрики после «стрижки». Наглядно видно, как показатели на тесте зависят от показателей при обучении. 9). Хорошо, что ещё больше улучшили результат для деревянных моделей. Вообще, все связки «модель-данные» индивидуальны, иногда более простые интерпретируемые алгоритмы показывают более высокие метрики, нежели сложные и перегруженные. Именно поэтому принято для любой новой задачи начинать с тестирования простых методов, и только затем переходить к сложным. Здесь ещё играет роль затратность по ресурсам и временной отработке моделей - в реальных компаниях и бизнесе это играет важную роль. 10). Тоже нет замечаний. 11). Датасет составлен правильно, предсказание тоже верное. Здорово, что сделали глубокие выводы и прописали рассуждения и ход работы в отдельных MarkDown-ячейках. Рекомендации: Можно тестировать и подбирать одновременно несколько гиперпараметров, для этого можно воспользоваться GridSearchCV (работает примерно как последовательность циклов, но позволяет более лаконично всё записывать в коде). Также есть библиотека optuna, она тоже делает подбор, но более грамотно и быстро – сразу прекращая процесс при обнаружении неоптимального «направления» значений гиперпараметра. С ними Вы подробнее познакомитесь в следующих модулях. Спасибо за выполненное задание! Если возникнут вопросы, мы всегда рады ответить на Ваши вопросы в пачке, пишите в канал 04\_ml\_03. Постараемся помочь разобраться со всеми моментами, которые вызывают трудности. Удачи в обучении!**