Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Департамент анализа данных, принятия решений и финансовых технологий

Пояснительная записка к курсовой работе

по дисциплине “Технологии обработки больших данных”

на тему:

«Машинное обучение в задачах кредитного скоринга»

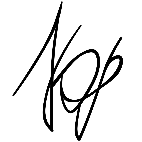
Направление подготовки 01.03.02 «Прикладная математика и информатика»

Профиль «Анализ данных и принятие решений в экономике и финансах»

Выполнил:

студент группы ПМ18-4 Факультета информационных технологий и анализа больших данных

Кайль Илья Олегович



Научный руководитель:

Профессор Чистов Дмитрий Владимирович

Москва – 2021 г.

Оглавление

[Введение 3](#_Toc58962425)

[1. Предварительный анализ данных 4](#_Toc58962426)

[1.1 Описание датасета. 4](#_Toc58962427)

[1.2 Предобработка данных 5](#_Toc58962428)

[2. Работа с моделями 12](#_Toc58962429)

[2.1 Модели классификации 15](#_Toc58962430)

[2.2 Сравнение моделей, выбор лучшей модели 17](#_Toc58962434)

[3. Улучшение выбранной модели. 18](#_Toc58962435)

[3.1 Стандатизация 18](#_Toc58962436)

[3.2 Кросс-валидация 18](#_Toc58962437)

[3.3 Грид сеч. 19](#_Toc58962438)

[3.4 Интерпретация молели 21](#_Toc58962439)

[Заключение 25](#_Toc58962440)

[Список литературы 26](#_Toc58962441)

[Приложение 1 27](#_Toc58962442)

## Введение

Цель данной работы является прогнозирование дефолта клиента на основе данных о его прошлой финансовой активности и иных данных. Использование моделей машинного обучения при решение данной задачи является актуальным и применимым, так как на данный момент крупные банки уже используют модели МО в виде инструментария для принятия решений, организуют соревнования с призовыми местами, с целью найти высококлассных специалистов и потенциального применения части или всего алгоритма победителей.

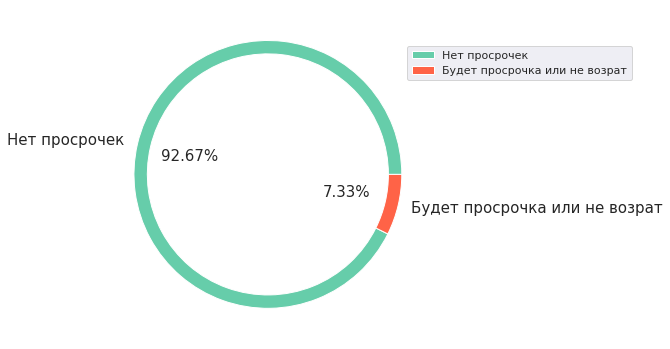
Для точного прогнзирования необходимо провести следующие действия, которые будут являться планом работы:

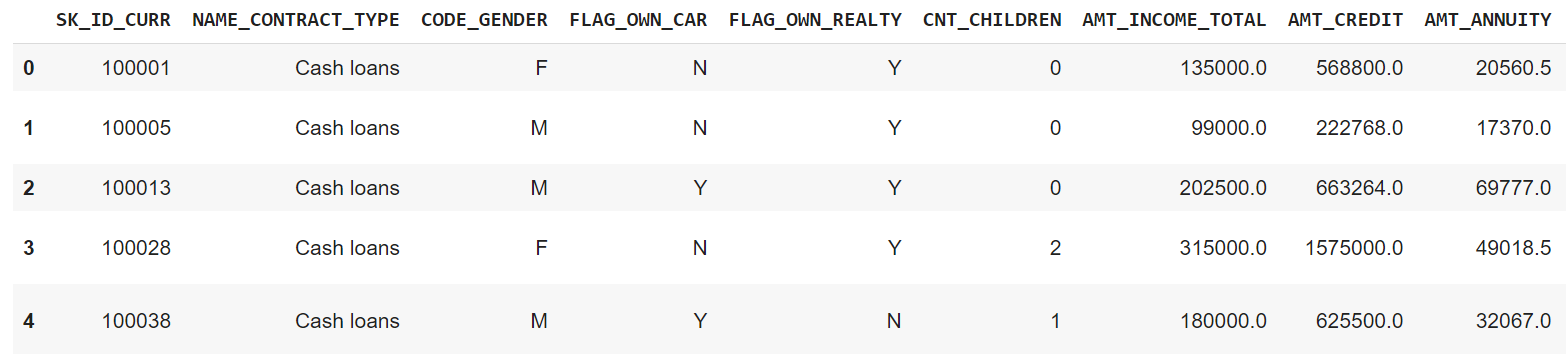
* Предварительный анализ данных, исключение признаков из исследования, а также работа с пропущенными значения;
* Выявление и работа с аномальными значениями и выбросами;
* Построение выбранных моделей классификации;
* Выбор лучшей из рассмотренных моделей;
* Улучшение предсказательной способности выбранной модели при помощи различных методов и проверка готовой модели на репрезентативность;
* Интерпретация настроенный модели

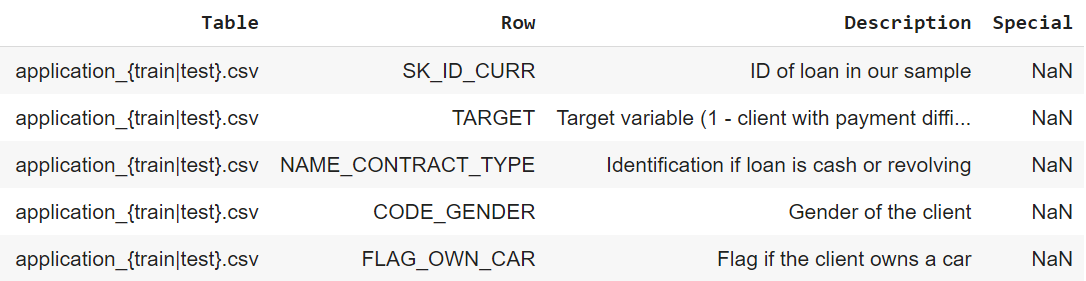
## Предварительный анализ данных.

### Описание данных.

Я использовал данные, предоставленные банком Home credit для соревнования на Kaggle. В рамках курсовой работы я решил использовать только 3 таблицы из 10 предоставленных банков.

«application\_train.csv» - тренировочные данные, на которых будет происходить обучение и оценка предсказательной способности модели. Таблица содержит 307 511 строк и 122 признаков. Среди таких признаков, как CNT\_CHILDREN, AMT\_INCOME\_TOTAL, AMT\_CREDIT и NAME\_INCOME\_TYPE есть целевая фича TARGET, которая содержит значения 1 - клиент с трудностями оплаты: у него/нее была просрочка платежа более чем на X дней по крайней мере по одному из первых Y платежей по кредиту в нашей выборке, 0 – отсутствие проблем по платежам. Стоит сразу отметить, что есть несбалансированность классов.

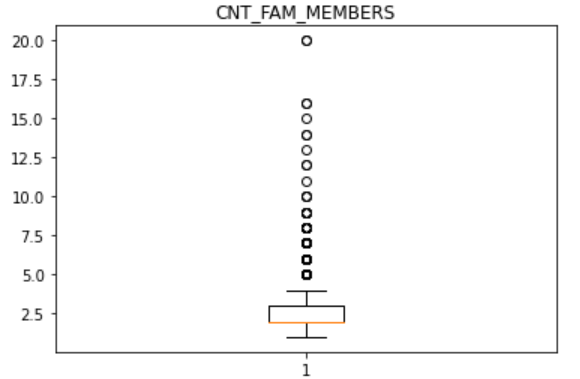
 «application\_test.csv» - тестовые данные, для которых я буду прогнозировать дефолт или отсутствие проблем по платежам. Отличие от «application\_train.csv» заключается в количестве строк, тут их 48 745 и в отсутствие целевой переменной. Столбцы такие же по смыслу, как и в train (для удобства, я так буду называть «application\_train.csv»).

 «HomeCredit\_columns\_description.csv» - описание каждого столбца для всех таблиц.

### 1.2 Предобработка данных.

В начале я удалил дубликаты, так как они не несут ценности для предсказательной способности модели. Затем, я решил удалить из данных столбцы, которое содержат более 50% пропусков. Но по-прежнему, остаются столбцы, которые содержат пропуски. Это не есть хорошо, так как многие модели машинного обучения не принимают на вход данные с пропущенными значениями. Необходимо заполнить данные признаки значениями или удалить строки, содержащие null. По итогу удалены строки с пропущенными значениями для всех числовых признаков, а для всех категориальных признаков заполнил самым популярным значением.

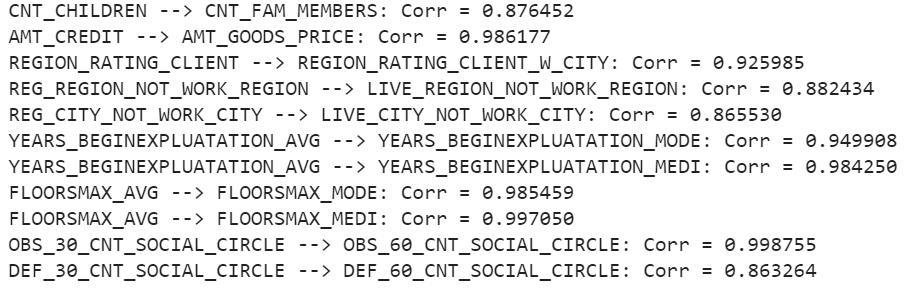
Далее я решил точечно удалять некоторые признаки. DAYS\_ID\_PUBLISH, представляет информацию о том, за сколько дней до подачи заявления клиент изменил документ удостоверяющий личность. Даты смены паспорта могут хранить инсайды, а в частности количество дат смены документа минус количества дат смены паспорта, установленных законодательно. Возможно, заемщик, который терял паспорт более N раза является не надежным, так как не серьезно относится к документам, и я могу предположить, что отношение к срокам погашения и в целом к возврату заемных средств тоже не серьезное. Однако, он мог восстанавливать паспорт и по другим причинам, возможно, есть клиенты, которые теряли документ не по своей вине. Нужны данные и проводить исследование, пока это только гипотеза. В данном случае тут просто 1 дата и с высоты своего опыта, я не вижу смысла оставлять данный признак в исследование. Признак SK\_ID\_CURR является id кредита. Может показаться, что данный столбец можно воспринимать как id заемщика. Однако, клиент мог ранее брать (рассматриваю train) не один кредит, и у каждого кредита свой id. Информации о том, какой заемщик брал кредит с SK\_ID\_CURR, например 227 у меня нет, следовательно, каждый SK\_ID\_CURR - просто уникальное значение.

 Затем, я перешел к этапу обнаружению аномальных значений и выбросов у числовых фичей и работе с ними. Аномальные значения и выбросы являются отклонением, которые не укладываются в общую модель поведения явления. Они мешают алгоритмам построить хорошее решающие правило. Я решил собрать список числовых столбцов для чистки аномальных значений и выбросов, а затем, основываясь на ниже указанных предположениях и фактах стал исключать из данного списка признаки, для которых я не буду проводить чистку. При помощи boxplot я проводил поиск фичей, содержащие аномалии или выбросы. Продемонстрирую один из построенных графиков:

Идентификация искомых значение, при помощи данной визуализация происходит путем отображения на графике точками значения, которые меньше, чем Q1-1.5\*IQR и больше, чем Q3+1.5\*IQR. Где Q1 – квартиль, равный 25-ому процентилю, Q3 – квартиль, равный 75-ому процентилю, а IQR – это разница между Q3 и Q1. Я написал функцию, которая будет удалять строки в зависимости от того, какого типа значений нелюходимо избавиться. Для аномалий диапазон остается прежним [Q1-1.5\*IQR; Q3+1.5\*IQR], то есть значения, которые выходят за данный интервал – являются аномальными, а для выбросов [Q1-3\*IQR; Q3+3\*IQR].

При рассмотрении boxplots было выявлено, что признак DAYS\_BIRH (количество дней от рождения до обращения с заявкой на выдачу кредита в банк) не нуждается в чистке. Признаки EXT\_SOURCE\_2 и EXT\_SOURCE\_2 тоже не будут подвержены чистки, хотя содержат небольшие отклонения. Значения данных фич – это оценка заемщика из внешних источников (компаний, которые специализируются на оценки клиента для банков). Значения ∈ (0;1). Я предполагаю, что они окажут огромный вклад в предсказание целевой фичи, так как банки покупали эту информацию, следовательно, считают данную оценку клиента полезной, и хотелось бы дать модели выявить лучшую связь между фичами и целевой переменной, не сокращая данную информацию (данные столбцы). Так же, я решил исключить признаки из списка исследуемых на аномальные значения столбцы, которые содержат всего 2-3 уникальных значения.

Закончив с отбором значений, я перешел к чистке. Запустив функцию для очистки столбцов, я обнаружил, что уже после чистки 14-ого признака таблица стало пустой. Моя задача заключалась в том, чтобы найти баланс между качеством чистки и сохранению объема данных, поэтому я принял решения удалить все значения, которые выходят за порог не аномальности для 7 столбцов.

 Затем, я перешел к этапу поиска сильно коррелирующих столбцов и исключения из исследования таковых. Сильная корреляция признаков между собой говорит о том, на сколько один столбец описывается другим. Стоит удалить один из парных столбцов, чтобы снизить избыточность данных, что не есть хорошо для моделей машинного обучения. Я ввел порог 0.8 и исключил признаки, значения корреляции которых больше по модулю.

Но на этом не все. Так же, для числовых признаков можно и нужно применить стандартизацию или нормализацию. Я не стал делать этого на данном этапе работы, так как хотел зафиксировать с помощью метрик результат данного преобразования. Однако, я это сделаю позже и стоит описать преобразование сейчас. Стандартизация (Standartization, Z-score Normalization) – техника преобразования числовых значений признака, адаптирующая ее с разными диапазонами значений к моделям машинного обучения.

*;*

Где - стандартизованный элемент признака, - исходный элемент, - среднее арифметическое, а – стандартное отклонение.

Также, можно применить методы понижения размерности данных, что не редко хорошо сказывается на предсказательной способности модели, однако, при применение данных методов порой теряется возможность интерпретировать модель, а в задаче кредитного скоринга – это важно. На момент написания работы я не обладал исчерпывающими объемом информации о подобных алгоритмах, поэтому, при написание дипломной работы, которая будет продолжение этой, я обязательно применю один из методов.

Далее, необходимо преобразовать категориальные признаки в числовые значения. Необходимость данного действия обуславливается тем, что многие модели не принимают на вход фичи, которые хранят внутри значения не типа float или int. Например, это можно сделать при помощи алгоритма LabelEncoder. В результате его применения каждому уникальному текстовому значению присваивается число, при этом исходный столбец сохраняется без создания новых. Также, есть алгоритм OneHotEncoder, который для каждого уникального значения в столбце создаёт новый столбец, присваивая значение внутри нового 1, если уникальное значение встречалось на данной строчке, а 0 если нет. И в результате применения алгоритма для каждого столбца удаляется исходный и создаются новые в количестве N-1, где N-число уникальных значений. Создание не ровно N столбцов объясню на примере пола. Столбец содержит 3 уникальных значения: мужчина, женщина и гендерфлюидный вертосексуал. Для каждого уникального значения нет необходимости создавать новый столбец исходя из логики авторов алгоритма: если человек не женщина и не мужчина, то значит это нечто Другое, и забивать новым столбцом с обозначение этого другого нет смысла. Анализируя, к какому признаку применить тот или иной алгоритм я исходил из того, что, применив OneHotEncoder для всех признаков, я создам почти сотню новых флаговых фичей. Так как в машинном обучение есть задачи понижения размерности данных я подумал, что не стоит бездумно насыщать данные для моделей и стоит проанализировать каждый столбец. Например, признаки, которые можно описать как больше или меньше, широкий или узкий, длинный или короткий можно преобразовать при помощи LabelEncoder, так как модель правильно поймет числовые значения в рамках одного признака, которые были получение путем: большой – 3, средний – 2, маленький – 1. А если взять категориальный столбец, который хранит категориальные значения сферы работы человека: кулинария, логистика, медицина, IT и т.д., то будет неправильно сравнивать их, например, на предмет того, кто из данной области более платежеспособный клиент и присваивать наибольшую или наименьшую оценку области, так как повар может зарабатывать сотни тысяч, работая в самом известном ресторане города, а может работать в обычном кафе.

Аналогично и для других сфер. По итогу, столбцов, которые стоило бы преобразовать при помощи LabelEncoder я не обнаружил. Поэтому, я преобразовал все столбцы при помощи LabelEncoder, создав около 80 новых столбцов. Таким образов увеличив количество признаков в 2 раза.

В конце разобью train на тренировочную выборку, на которой буду обучать модель и тестовую, на которой буду оценивать предсказательную способность модели. Пока что, разобью в соотношение 70:30 и установлю параметр разбиения random\_state = 42, чтобы зафиксировать определенное разбиение.

На этом предобработка данных закончена.

## Работа с моделями.

Прежде чем провести тренировку моделей и выбрать лучшую, необходимо выбрать метрики, на которых будет происходить оценка модели. Так как в данной работе необходимо предсказать бинарное значение целевой фичи, я буду решать задачу классификации. Конечно, можно решать задачу предсказания вероятности, используя регрессионные модели и выставить самостоятельно порог, при котором можно считать, что будет проблема с клиентом, однако, я не решил пойти по данному пути решения задачи.

Самые распространенные метрики классификации это Accuracy, AUC-ROC, Precision Recall и F1-Score.

Ранее я зафиксировал несбалансированность классов, по этой причине я не буду использовать Accuracy, так как Accuracy – это доля правильных ответов алгоритма, без учета предсказательной способности каждого класса. Исследуемые данные имеют соотношение 93% и 7% мажорного и минорного класса. Можно сделать предположение, что практически любая модель будет иметь низкую предсказательную способность для минорного класса, а Accuracy будет иметь значение 0.8-0.9, что будет вводить в заблуждение о качестве модели. Минорный класс – клиенты, с которыми будут проблемы по платежам. Банку важнее предсказать клиента и убытки по сумме выдачи заемных средств и упущенную прибыль в виде процентов, чем не выдать кредит по какой-либо заявке и лишиться прибыли по процентам.

В случае дисбаланса AUC-ROC не является панацеей, однако в качестве дополнительной проверки данная метрика не будет лучшей. AUC-ROC – площадь под кривой ошибок. Данная прямая представляет из себя линию от (0,0) до (1,1) в координатах True Positive Rate (TPR) и False Positive Rate (FPR):

;

;

где, – доля объектов, предсказаны как положительный класс и являются положительными; – доля объект предсказаны как отрицательный класс, но не являются таковым; – доля объектов, предсказаны как положительный класс, но являются отрицательным; – доля объектов, предсказаны как отрицательный класс и являются отрицательным. Продемонстрирую матрицу ошибок для лучшего понимания:

|  |  |  |
| --- | --- | --- |
|  | =1 | = 0 |
| =1 | True Positive (TP) | False Positive (FP) |
| =0 | False Negative (FN) | True Negative (TN) |

Где - истинная метка класса, а – предсказанная моделью метка.

Самыми важными в данной работе являются Precision и Recall.

Precision – можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными. Цель Precision классифицировать все Positive семплы как Positive, не допуская ложных определений Negative как Positive (распознанных правильно и неправильно). Другими словами, Precision -демонстрирует способность отличать этот класс от других классов. Когда модель делает много неверных Positive классификаций, это увеличивает знаменатель и снижает Precision.

Recall – показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм. Когда модель правильно классифицирует все Positive выборки, recall будет 100%, даже если все представители класса Negative были ошибочно определены как Positive.

F1-Score— среднее гармоническое precision и recall. Данная метрика показывает, насколько точен классификатор (сколько экземпляров он классифицирует правильно), а также насколько он устойчив (он не пропускает значительное количество экземпляров). Имея в своем распоряжении подобный механизм оценки будет гораздо проще принять решение о том являются ли изменения в алгоритме в лучшую сторону или нет. В реальной жизни максимальная точность и полнота не достижимы одновременно и приходится искать некий баланс. Поэтому, хотелось бы иметь метрику, которая объединяла бы в себе информацию о точности и полноте алгоритма. Существует несколько различных способов объединить precision и recall в агрегированный критерий качества. F-мера (в общем случае ) — среднее гармоническое Precision и Recall:

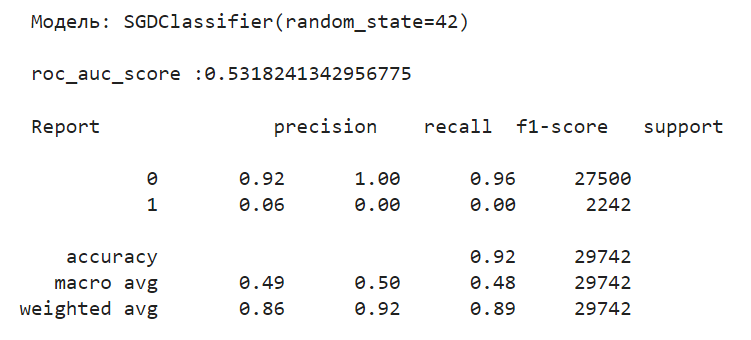
в данном случае определяет вес точности в метрике, и при = 1 это среднее гармоническое (с множителем 2, чтобы в случае precision = 1 и recall = 1 иметь F1 = 1) F-мера достигает максимума при полноте и точности, равными единице, и близка к нулю, если один из аргументов близок к нулю.

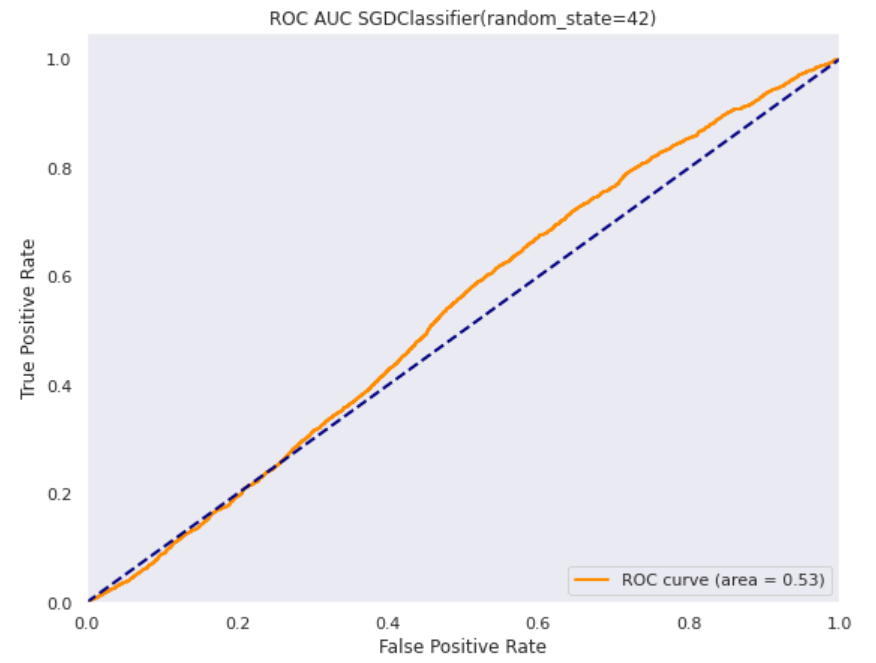
### 2.1 Модели классификации.

В рамках данной задачи я выбрал 10 моделей классификаций. На данном этапе развития как специалиста, я не могу жонглировать знаниями о каждой модели, и обоснованно взять на рассмотрение только 3 модели, которые лучше поймут данные и на практике и опыту людей, лучше подходят к области (как в моем случае - кредитного скоринга), а не 10, что сэкономит время. Однако, я хочу подготовить модель, которая точнее будет прогнозировать Y и придется обучать 10 моделей, потратив на это больше времени.

Список моделей для обучения с установленным для всех параметром random\_state=42: XGBClassifier, CatBoostClassifier, LGBMClassifier, RandomForestClassifier, KNeighborsClassifier, DecisionTreeClassifier, LogisticRegression, GaussianNB, SVC, MLPClassifier, GradientBoostingClassifier, RidgeClassifier и SGDClassifier.

Так как я ранее уже описывал уровень своей компетентности, я не считай честным и разумным расписывать по формулам каждую модель, не отдавай себе отчету почему расчеты в каждой модели являются применимыми в рамках моих данных, ведь в таком случае описание модели будет ради описания, а не моего обоснованного выбора. Так же по результату обучения и анализу метрик я выберу всего одну модель, без четкого понимания таковой, и в таком случае описание непригодных моделей будет служить лишь дополнительной справкой, которая никак не объяснит и не аргументируют ход моих действий. Однако, я опишу лучшую модель по результату тренировок. Не сочтите мой подход дерзким и девиантным, у меня есть желание узнать и понять исследуемую область, но, к сожалению, в рамках уделенного времени на курсовую работу у меня не хватило времени выполнить ее превосходно.

Для обучения моделей и выводу метрик я написал функцию, которая на ход получает данные, разбитые раннее для обучения и предсказания и список моделей. Так же, в результате работы происходит вывод графика площади AUC-ROC для каждой модели. Продемонстрирую ниже для 1 моделей результат выполнения функции:



### 2.2 Сравнение моделей, выбор лучшей модели.

Изучая значения полученных метрик, можно убедиться в том, что использование метрики Accuracy ошибочно, так как для всех моделей данное значение 0.92 - 0.93, вводя в заблуждение о хорошем качестве модели. Однако модель не обладает таковым качеством, так как есть несбалансированность классов и по классификационному отчету это можно подтвердить значениями метрик Recall, Precision и F1 для минорного класса редко принимающие приемлемые значения (только для 3 моделей), а чаще всего стремящиеся к 0.15 и во все к 0. Однако, данные метрики показывают отличные значения для мажорного класса. Получается, что модели практически без ошибочно определяют 1 из классов, при этом практически не распознавая другой, что не есть хорошо, так как в результате модель обучена почти всегда идентифицировать объекты как мажорный класса.

Стоит выделить 3 модели, которые показали лучший и эквивалентный результат среди всех: XGBClassifier, CatBoostClassifier и GradientBoostingClassifier. Стоит отметить, что данные модели объединяет схожий метод машинного обучения - градиентный бустинг. Лучшими их определяет значение Precision 0.5 и значение AUC-ROC 0.70-0.74.

Я решил дальше работать с моделью CatBoostClassifier, так как она поддерживает практически полезные визуализации во время обучения и других процедур, разработчики данной модели публикуют лекции, в которых рассказывают о параметрах модели, методах и практических советах по работе с ней, что поможет с ее тюнингом и лучшим понимаем отчетов и значений параметров. Также, данная модель поддерживает визуализации для интерпретирования моделей shap, которые являются сейчас актуальными в профессии.

## Улучшение выбранной модели.

На данном этапе работы необходимо стандартизировать данные, попытаться побороть несбалансированность классов, провести кросс- валидацию, грид сеч и тюнинг модели.

### Стандартизация и присвоение весов.

Ранее я уже описал стандартизацию. Также, на этом этапе я воспользовался методом class\_weight из библиотеки sklearn для того, чтобы функции потерь присвоить минорному классу более высокое значение. Следовательно, потеря становится средневзвешенной величиной, где вес каждой выборки определяется class\_weight и соответствующим классом.

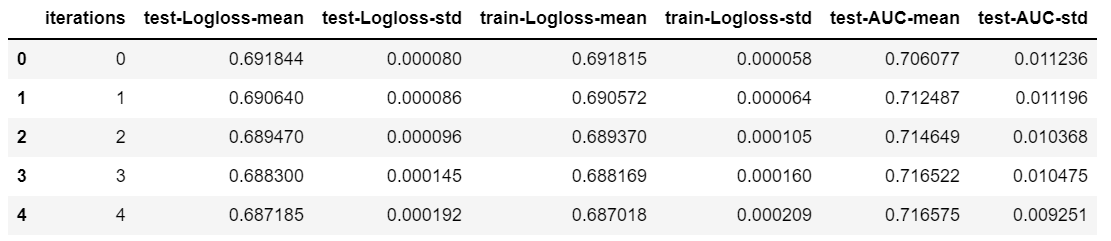
В результате проведения стандартизации и присвоения каждому классу вес (в параметре модели) значения AUC выросло на 0.002, Recall + ~ 0.07 (для минорного класса, который был ~ 0) и Precision остался ~ 0.45 (для минорного класса, который был ~ 0.45). Также, 2 последние метрики снизились в рамках ~ 0.1 - 0.2 для мажорного класса.

Однако, проблема дисбаланса не решена применением одним из методов борьбы. В рамках продолжения данной работы в дипломе я рассмотрю иные методы.

### Кросс-валидация.

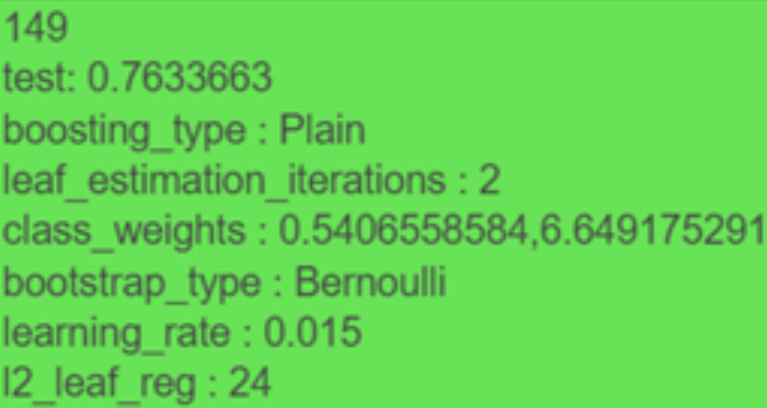
Кросс-валидация - метод оценки аналитической модели и её поведения на независимых данных. При оценке модели имеющиеся в наличии данные разбиваются на k частей. Затем на k−1 частях данных производится обучение модели, а оставшаяся часть данных используется для тестирования. Процедура повторяется k раз. В итоге каждая из k частей данных используется для тестирования. В результате получается оценка эффективности выбранной модели с наиболее равномерным использованием имеющихся данных.

Я установил k = 5. Результат перекрестной проверки:

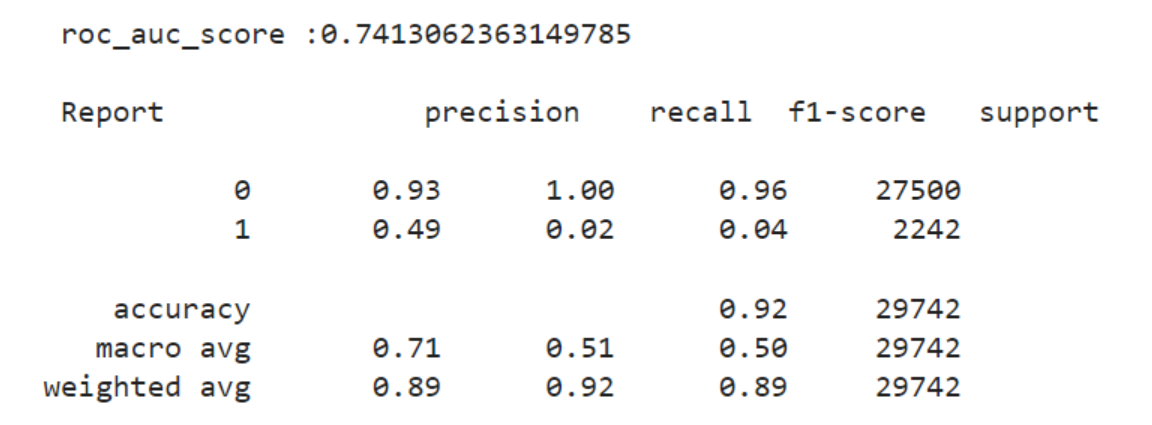


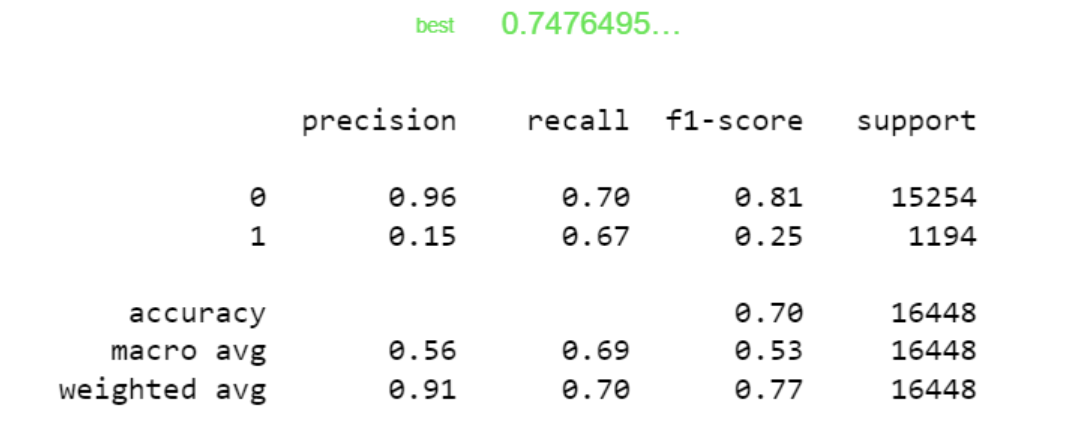
В результате осознал ошибку того, что данную процедуру стоило провести после тюнинга и грид сеча. Среднее значение AUC модели удовлетворительное, без учетов ее тюнинга и проведения грид сеча.

### Грид сеч.

Грид сеч – перебор параметров для выявление наиболее подходящих для модели. Перед запуском данного алгоритма я установил параметры модели перебора и значения, по которым я хотел сделать перебор. В результате лучшими параметрами оказались:

По итогу, до применения всех этап в данной главе и установки лучших моделей классификационный репорт был:

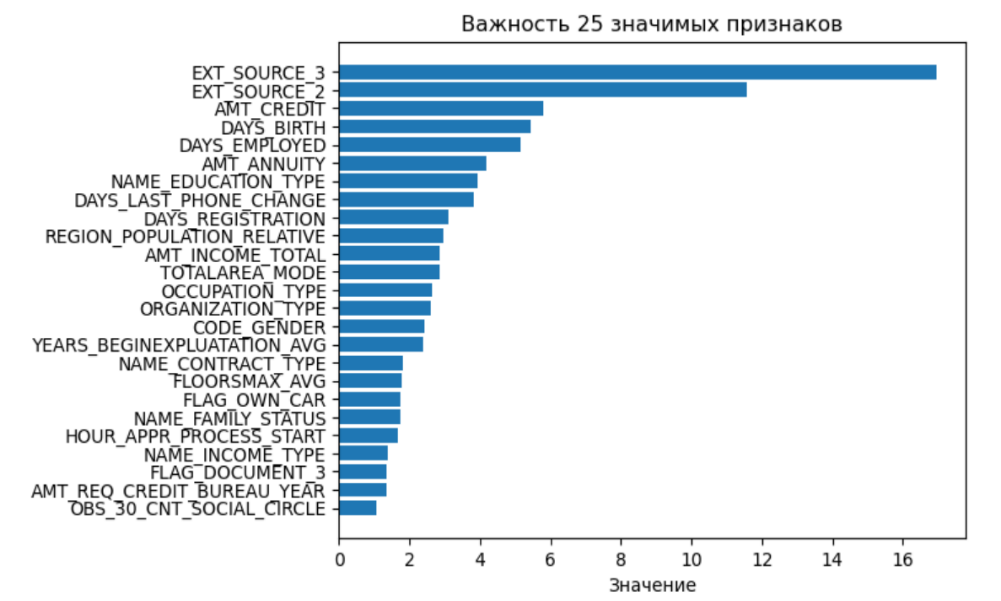


А после всех описанных действий стал:

В результате Precision - доля объектов, названных классификатором положительными и при этом действительно являющимися положительными уменьшилось для минорного класса, а Recall - доля объектов положительного класса из всех объектов положительного класса нашел алгоритм вырос, также для минорного класса. Также, упала предсказательная способность для мажорного класса. Также, удалось поднять значение метрики F1-Score для минорного класса.

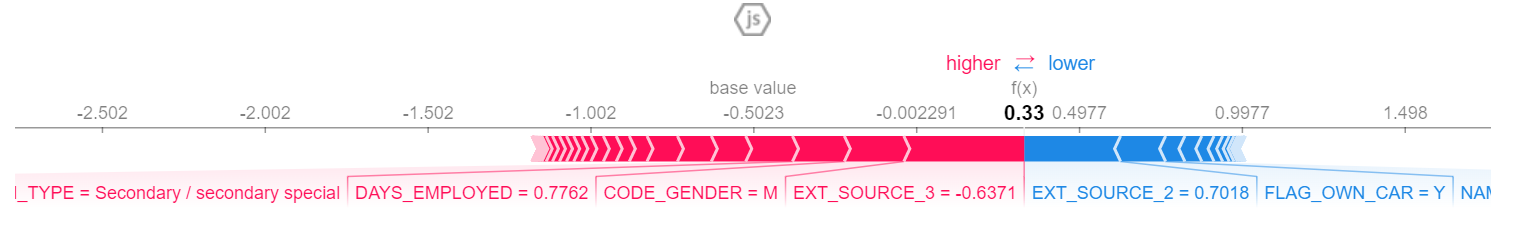
### Интерпретация модели

Необходимость данного этапа обуславливается отбором действительно необходимых фичей для дальнейшего обучения и исключением шумовых. Также необходимостью объяснить заказчику/начальнику/руководителю/и т.д. как модель понимает данные, почему выделяет одни признаке на фоне остальных и некую понятную констатацию закономерностей.

Перейду к оценки фичей при помощи PredictionValuesChange. Для каждого объекта PredictionValuesChange показывает, насколько в среднем изменяется прогноз при изменении значения объекта. Чем больше значение важности, тем больше в среднем изменение значения прогноза, если эта функция будет изменена. Значения важности признаков нормализованы так, чтобы сумма важностей всех функций была равна 100. Выведу 25 признаков, оказывающие наибольшее влияние на предсказание, согласно расчетам данным способом:

Мне бы не хотелось утверждать, что данные признаки действительно являются важными с учетом всего сказанного и продемострированного о предсказательной способности модели. Также, я бы не стал основываться только на данном методе расчета (предположим, что моя модель является хорошо обученной), а взглянул бы на совокупность результатов разных методов подсчета важности признаков на таргет.

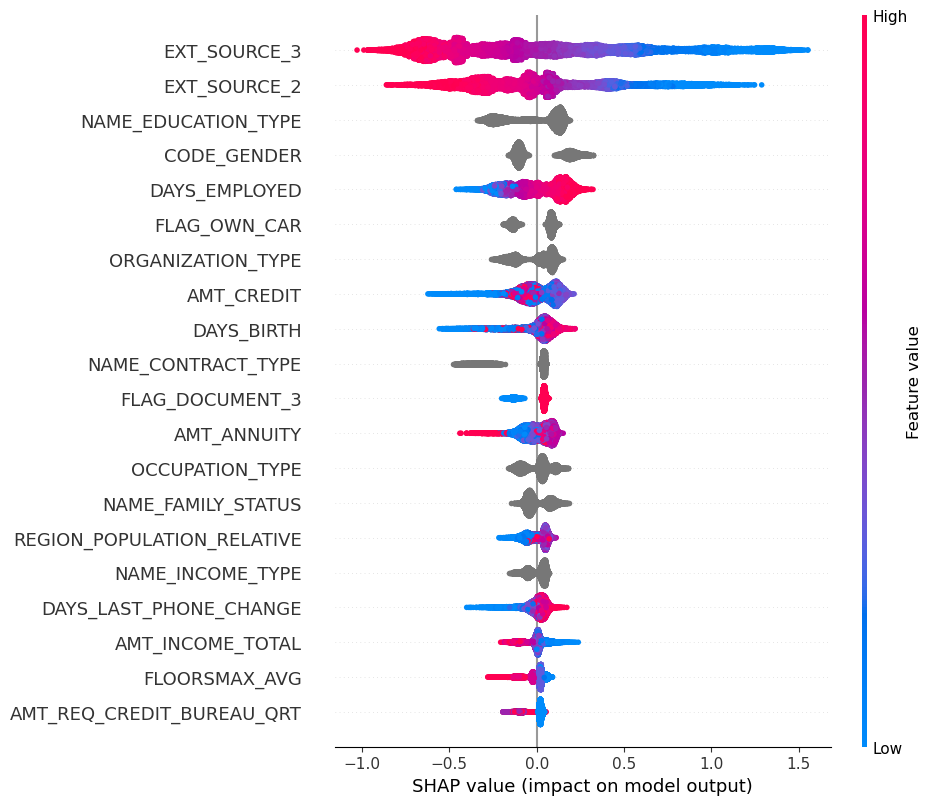
Shap - это теоретико-игровой подход для объяснения результатов любой модели машинного обучения. Он связывает оптимальное распределение кредитов с локальными объяснениями с использованием классических значений Шепли из теории игр и связанных с ними расширений.

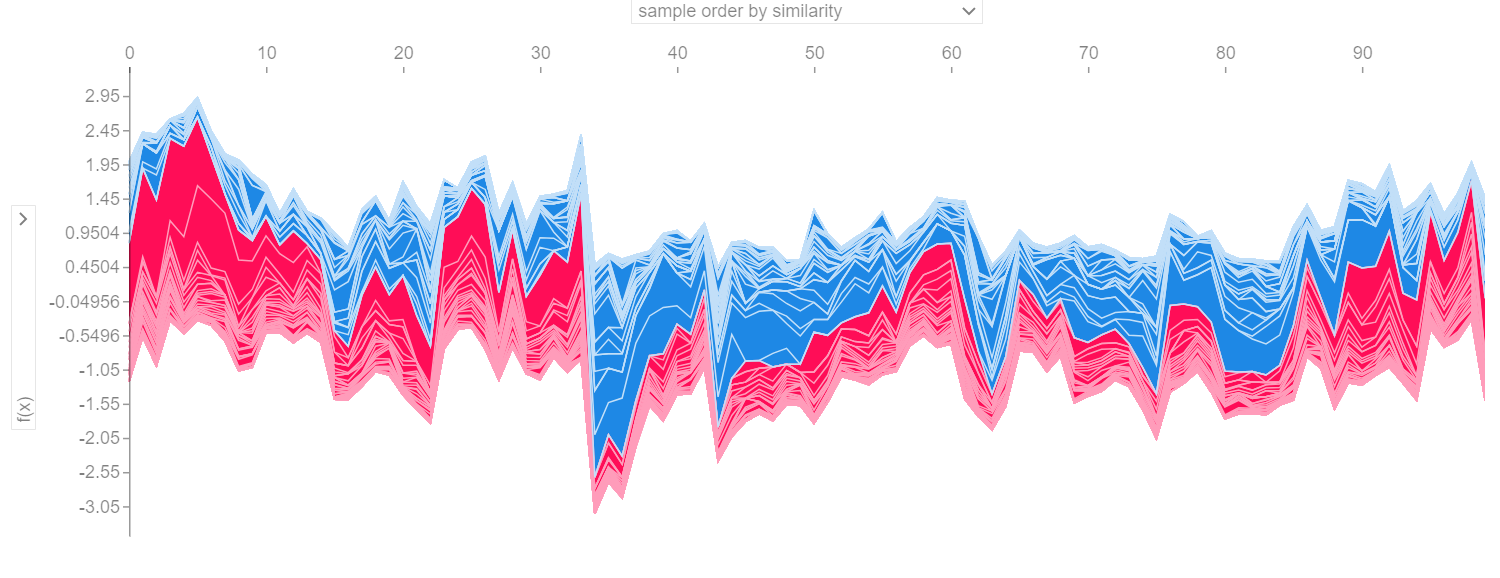
Чтобы понять почему какой-то объект (заемщик) был предсказан таким образом, можно посмотреть вклад других фичей в предсказание выбранного:

Красные признаки - двигают предсказание в положительную сторону, то есть они увеличивают вероятность того, что для данного объекта будет предсказан невозврат средств / проблемы по платежам. Синие фичи - снижают вероятность данного события.

Данный график можно интерпретировать так: для заемщика признаки EXT\_SOURCE\_3, CODE\_GENDER и DAYS\_EMPLOYED вносят вклад в предсказание дефолта данного объекта, а признаки EXT\_SOURCE\_3, FLAG\_OWN\_CAR в предсказание того, что объект надежный.

Чтобы получить представление о том, какие признаки наиболее важны модели, можно построить значения SHAP для каждого признака с каждым объектом. График состоит из точек. Каждая точка - обучающий объект выборки. Для точек, расположенных справа от вертикальной черты, вклад от данной фичи положительный, то есть они увеличивают вероятность того, что для данного объекта будет предсказан не невозврат средств/ проблемы по платежам. Чем шире полоса, чем больший общий вклад внесла фича для совокупности объектов. Цвет представляет значение фичи (красный высокое, синий низкое). Серый цвет представляет собой категориальные значения, которые не могут быть увеличены или уменьшены. Но специалисты по сопровождению пакетов работают над этим (пока что это единственное объяснение, которое я смог найти).

Как уже было отмечено, стоит оценивать клад признаков в совокупности разных подходов расчеты меры важности. Сравнивая значимые признаки при расчете PredictionValuesChange и Shap можно заметить, что в порядке по убыванию важности при обоих расчетах важными признаками чаще всего являются одни и те же столбцы.

 Чтобы посмотреть влияние столбцов не для 1 объекта, а для N, можно построить следующий график (суть цветов, как и для 1 графика визуализаций shap).

Также, стоит интерпретировать модель при помощи LIME и других методов, но это уже будет рассмотрено в рамках дипломной работы. Сейчас, для "плохой" модели, углубленная интерпретация, как мне кажется, не столь важна.

## Заключение

В результате работы был выполнен поставленный план, однако мне не удалось увеличить качество прогнозируемости модели по причине такого, что мне не удалось побороть несбалансированность, некоторые этапы работы пост факту анализа были выполнены некорректно или проигнорированы вовсе. Дипломная работа будет продолжением данной курсовой работы, поэтому одной из целей дипломной работы будет переосмысление к подходу выбора методов текущего плана, с целью подготовить качественную модель.

## Список литературы

1) Метрики в задач машинного обучения [Электронный ресурс]: // URL: https://habr.com/ru/company/ods/blog/328372/(Дата обращения: 25.12.2021).

2) Мастер класс. Решение задач классификации при помощи CatBoost -Никита Дмитриев обучения: Компьютерные технологии [Официальный аккаунт на YouTube]: // URL: https://youtu.be/xl1fwCza9C8 (Дата обращения: 27.11.2020).

3). Документация библиотеки shap. Примеры: [Электронный ресурс]// URL: https://github.com/slundberg/shap (Дата обращения: 23.12.2021).

4). Библиотека scikit-learn: [Электронный ресурс]// URL: https://scikit-learn.org/stable/ (Дата обращения: 28.11.2020).

5) Валидация моделей машинного обучения: [Электронный ресурс]// URL: https://habr.com/ru/company/glowbyte/blog/569970/ (Дата обращения: 28.12.2021).

## Приложение 1

* - репозиторий, содержащий программный код курсовой работы, а также используемый датасеты.