Претерпевает сокращение доля статьи «Фонды переоценки статей баланса». Заметную, расширяющуюся долю занимает статья «Резервный фонд».

И так по результатам анализа структуры баланса Банка можно сделать следующие выводы:

1. Банк в первую очередь ориентирован на работу с клиентом: на рисунках 3.1 и 3.2 доли статей связанный с выдачей кредитов и получением депозита – самые заметные;
2. Кроме активов связанных с работой с клиентом банк держит большие доли ценных бумах и средств в банках, общая доля соответствующих категорий примерно постоянна, хотя между собой они претерпевают некоторые мутации;
3. Банк последние годы активно сокращает долю средств других банков в обязательствах;
4. Доля собственного капитала остается из года в год почти неизменной, хотя внутри нарастает влияние накопленной прибыли, что говорит о неплохом положении дел в последние годы.

2.2 Начальный анализ наблюденных данных

Сразу укажем, что все действия связанные с вычислениями и обработкой информации мы производили на языке программирования python3 – одном из самых популярных и востребованных, на сегодняшний день, инструментов для математического моделирования и обработки данных. В работе использованы библиотеки-расширения pandas [7], numpy [8], scipy [9], sklearn [10], pytorch [11]. Весь исходный код в формате jupyter notebook или исполняемых «.py» файлов, результаты его выполнения в необработанном виде и даже текст этой работы можно найти на предварительно созданном репозитории[1] в сети интернет.

В связи с коммерческой тайной мы не можем предоставить данные использованные для построения модели в открытый доступ, приходиться принять, вызванную этим фактом, непрозрачность описываемого исследования. Тем не менее, мы на словах и промежуточных результатах вычислений постараемся по максимуму раскрыть полный цикл разработки модели – от «сырой» таблицы показателей и до валидации готовой модели.

Этот раздел посвящен анализу полученной из баз данных банка информации и, на его основании, обоснования и применения решений связанных с преобразованием данных способствующих дальнейшему успеху при моделировании.

Для преобразования данных зачастую пишут отдельную программу со своими настройками, так как по результатам моделирования может потребоваться подгонка еще и параметров предварительной обработки данных – в этом случае следует просто поменять некоторые настройки процессинга данных и забрать по сути новый набор данных. Такую программу в западных источниках называют «pipeline» (пайплайн), мы будем называть конвейер

[1] https://github.com/Dranikf/diplom_project

данных. Учитывая, что конвейер данных собирается под конкретную структуру данных и то, что нет возможности предоставить данные в свободный доступ, не вижу никакого смысла, в техническом представлении этой программы в работе. Периодически лишь будут мелькать обобщенные методы для конкретных преобразований и представления результатов вычислений.

Та часть практики, которая связана с аналитикой и подготовкой данных, может быть найдена в папке «data_processing» названного репозитория. Особое внимание следует обратить на файл «processing1.ipynb», его можно открыть прямо в браузере.

Изначально входной файл идет в формате «.xlsx» – его, при соблюдении внутри структуры как в таблице 1.1 можно в одну строку открыть с помощью метода «read_excel» библиотеки pandas. На выходе будет получен «padnas.DataFrame» – объект представляющий собой данные вида таблицы 1.1. Заметим, что большой объем данных из формата «.xlsx» загружается достаточно долго, потому его лучше преобразовать в формат «.csv», который значительно проще. Сделать это можно прямо внутри pandas используя метод «to_csv» объекта «padnas.DataFrame», важно в аргумент «index» передать значение «none» для того, чтобы не был сохранен еще и столбец с индексом который был автоматически создан при загрузке. После этой процедуры можно пользоваться созданным «.csv» файлом, загружая его функцией «read_csv» библиотеки «pandas».

И так, изначальный набор данных имеет 247062 записей и 45 столбцов-показателей. Для начала нужно разобраться с составом столбцов, их типом данных и структурой. Все эти операции можно провести используя только возможности pandas, но удобнее, чтобы это можно было сделать в одну строку кода. Для решения такой задачи создана функция языка программирования python «get_data_frame_settings» исходный код представлен на листинге В.1. Применив ее к объекту с данными, на выходе будет получен, также «pandas.DataFrame». Для наших данных результат представлен в таблице Д.1.

Первый столбец содержит название показателя. Второй столбец посвящен типу данных, конечно в pandas типы данных другие – но в листинге В.1 объявлен словарь «types_natural_names» в котором реальным названиям типов данных pandas поставлены в соответствие русскоязычные названия. Можно заметить, что перечислены не все типы данных pandas и те которые не упомянуты будут отображаться так, как они бы выглядели в pandas, в случае необходимости можно добавить свой пункт в этот словарь. И так, у нас в таблице имеются данные следующих типов – дата, целое число, действительное число и номинативная переменная.

Третий столбец содержит информацию о возможных значениях которые принимает соответствующий показатель. Для дат и чисел это будет интервал, для номинативных показателей перечисленные через запятую уровни наблюденные в этом столбце. Сразу заметим, что показатели «Адрес проживания - Населенный пункт», «Вид деятельности по ОКЭД», «Кредитный продукт» и «Место работы» принимали настолько много уровней, что все их выписать в эту работу не

представлялось возможным, из соображений оформления, потому они были заменены.

Четвертый столбец, для номинативных переменных содержит число уровней которые они принимают.

Отдельную сложность в подготовке и анализе данных представляют ячейки с пропущенными значениями, в последнем столбце таблицы Д.1 выписаны количества таких наблюдений. Некоторые столбцы содержат до 95% пропусков.

Теперь, отталкиваясь от этой таблицы попробуем принять некоторые преобразования данных.

Особую ценность для нас составляет столбец «Дефолт», в нем содержится, целевая для нас информация – сколько дней, на момент среза, таблицы каждый контракт находиться в состоянии просрочки платежа. По указанию начальства, для целей моделирования, вышедшим в дефолт считается контракт, у которого в этом столбце наблюдается значение свыше 59 дней. Произведем перекодировку в новый столбец «*Y*» по правилу:

$$Y = \begin{cases} 1, \text{Дефолт} < 60; \\ 0, \text{в противном случае.} \end{cases}$$

Исходный столбец «Дефолт» подлежит удалению.

Выше уже упоминались столбцы «Вид деятельности по ОКЭД», «Кредитный продукт» и «Место работы». Было принято решение произвести их удаление. Все они содержат очень много уровней, что в условиях модели привело бы к колоссальному росту объема вычислений, хотя на решения модели это вряд ли как-либо повлияло – если некоторый уровень редко встречается, даже если все восхождения отметились как дефолт, нет оснований считать, что отловлена статистическая закономерность. Можно было бы попытать произвести укрупнение этих столбцов, но это было осложнено но названным ниже причинам.

Столбец «Вид деятельности по ОКЭД» идет в достаточно странном формате. Опираясь на общегосударственный классификатор видов экономической деятельности [12] удалось выяснить, что в уровнях присутствуют, как секции (буквенные обозначения), так и разделы, которые вообще говоря входят в секции. Непонятны причины такого разделения, потому столбец и было решено удалить.

Столбцы «Кредитный продукт» и «Место работы» подавались в строковом формате. Кончено, были «зацепки» для проведения некоторого парсинга, но из соображений экономии времени и не перспективности данных направлений работы было решено этого не делать.

Перейдем к показателю «овердрафт»: бинарный показатель который показывает является ли каждый конкретный кредит овердрафтом. Овердрафт – особый вид кредита, он предполагает открытие некоторого счета, в котором можно уйти в отрицательную сумму – только тогда появляется задолженность. Дело в том, что момент появления задолженности может не совпадать с

моментом, когда была подана заявка на открытие такого счета и, как следствие, этот вид кредита имеет особую специфику. Кончено было бы идеально, чтобы модель автоматически учитывала этот факт – но это может значительно осложнить процесс моделирования, потому пока модель будет построена только для случаев без овердрафта. Соответственно предполагается удаление всех наблюдений содержащих уровень «да» в рассматриваемом столбце и сам столбец вырождается, потому подлежит удалению.

Основная цель моделирования в данном случае – построение некоторого правила, которое поможет оценить способность каждого конкретного кредитополучателя к возврату долга. Потому, все показатели должны быть известны на момент выдачи кредита. Мы заметили, что исходной таблице присутствуют показатели которые не могут быть известны заранее: «Отношение факт срока к плановому при прекращении КД», «Причина прекращения договора» и «Дата фактического закрытия»; такие показатели также подлежат удалению.

Нередко показатель в чистой форме имеет слабую взаимосвязь с исследуемым явлением или не имеет взаимосвязи вовсе, но если к нему применить некоторые преобразования – то он может быть куда более полезен. Сейчас мы просто покажем процесс создания ряда показателей, а в следующем подразделе займемся выбором наилучших.

В таблице Д.1 сразу попадаются на глаза показатели связанные с годом выпуска автомобиля, притом они очень невнятно подписаны и мне так и не удалось выяснить, чем они между собой отличаются. Можно лишь выдвигать предположения результатом которых станет некоторое комбинирование этих показателей. Так были созданы показатели «Автомобиль год выпуска» который представлял собой просто самую позднюю дату из исходных трех, «Число авто» в котором подчитывалось количество непустых значений исходных трех, есть ли авто который принимал значение «да» если находился хотя бы одно непустое значение в исходных.

Продолжая разговор о показателях которые содержат дату, можно сказать, что показатели «Дата планируемого закрытия» и «Дата регистрации договора» не могут быть использованы в модели, так как уже наблюденные даты никогда больше не повторяться в новых заявках. Но вот их разность покажет предполагаемый срок кредита, с этой идеей был создан показатель «Срок кредита в днях». Исходные показатели были удалены из выборки.

Дальнейшим развитием этого показателя послужит его использование вместе с показателем «Сумма договора». Разделив показатель «Сумма договора» на показатель «Срок кредита в днях» получим показатель «Ежедневный платеж».

Модель не воспринимает дату, как формат, потому преобразуем его в число как количество дней от базовой даты – 8 декабря 1991 года.

Заметим, что среди показателей с очень большим числом уровней также оказался показатель «Адрес проживания - Населенный пункт», но он был выше удален подобно «Вид деятельности по ОКЭД», «Кредитный продукт» и «Место работы». Для его относительно легко оказалось провести парсинг. Можно было

выделить показатель «Столица», который принимал значение «да» если запись относилась к городу Минску и нет в противном случае. Аналогичные действия относительно легко было провести с областными центрами с результатом в виде бинарного показателя «Областной центр». Исходный показатель был заменен так что все наблюдения не относящиеся к одному из областных центров получили значение «нет информации». Прежде чем проводить описанные процедуры надо было обязательно подготовить исходный столбец – дело в том, что один и тот же уровень мог быть записан по разному в плане некоторых отдельных символов, например в выборке можно было найти 51 запись в которой был указан город «могилев» (с маленькой буквы и буквой «е» вместо «ё»), хотя большинство записей указаны по правилам – «Могилёв». Для того, чтобы обойти такую погрешность нужно все символы столбца перевести в нижний регистр и буквы «е» заменить на «ё».

По завершению этого этапа выборка уже несколько трансформировалась и это не может не отразиться на таблице Д.1. В таблице Д.2 представлена актуальная информация о показателях выборки.

Следующим шагом станет очистка выборки от выбросов. В основном, тут следует уделять внимание столбцу «Область допустимых значений» таблицы Д.2.

Сразу обратим внимание на показатели «Работа последнее место стаж лет», «Работа уровень дохода BYN», «Сумма договора» и «Ежедневный платеж», они по смыслу, очевидно, не могут содержать отрицательные значения, однако по проведенному исследованию получается, что есть записи с отрицательными числами. Очевидно, это некоторая ошибка заполнения, которую нам следует исправить. Будет неправильно удалять целую запись с ошибочным заполнением, так как по результатам анализа классифицирующей способности, проводимого далее, может быть удален целый столбец. Правильнее будет эти позиции заполнить пропусками, так информация, содержащаяся в прочих столбцах, будет сохранена.

Далее нам показались подозрительными размахи некоторых переменных: «Количество фактов просрочки по основному долгу», «Максимальное количество дней просрочки», «Общее количество запросов в КБ», «Сумма кредитных лимитов», «Сумма договора» и «Ежедневный платеж». Для быстрого исследования квантилей числовых показателей в pandas можно использовать следующий механизм – выделяя через оператор «[]» некоторый столбец получим объект типа «pandas.Series» у которого имеется метод «describe». В выводе названного метода будет информация о числе непропущенных наблюдений, среднем, стандартном отклонении, размахе, 25%, 75% персентилях и медиане. Для быстрого получения информации этой информации сразу по диапазону столбцов может быть использована функция «get_describes» нанесенная на листинге В.2. Так краткой записью:

```
get_describes(data[em_research_list]),
```

где data – pandas.DataFrame содержащий все исследуемые показатели;
em_research_list – список численных показателей.

Можно получить таблицу вида 2.1.

Таблица 2.1 – Описательные статистики столбцов с предполагаемыми выбросами

Статистика	Количество фактов просрочки по основному долгу	Максимальное количество дней просрочки	Общее количество запросов в КБ	Сумма кредитных лимитов	Сумма договора	Ежедневный платеж
Количество	77016,00	70482,00	104134,00	90858,00	236496,00	236495,00
Среднее	5,11	18,05	8,35	7719,67	5739,18	2,69
СКО	10,60	132,79	7,48	13222,99	13752,80	2,74
Мин.	0,00	0,00	0,00	0,00	0,00	0,00
25%	0,00	0,00	3,00	1187,16	600,00	1,14
50%	1,00	1,00	7,00	3800,00	1500,00	1,86
75%	5,00	8,00	11,00	8900,00	5000,00	3,35
Макс.	284,00	4471,00	222,00	678562,60	600000,00	196,24

Примечание – Источник: собственная разработка.

Особо важная для нас в данном случае информация содержится в стоках «75%» и «Макс.». Заметим, что в каждых из названных столбцов максимальное значение очень отдалено от 75-го персентиля, что, скорее всего, вызвано некоторыми исключительными случаями. Так как основная идея моделей на статистических данных – отловить центральную тенденцию, а наличие таких случаев в выборке может вести к смещению оценок параметров, то их обычно удаляют.

При отделении значений относящихся к выбросам можно пользоваться следующим правилом [13], выбросами считаются наблюдения лежащие вне интервала:

$$[x_{25} - 1{,}5(x_{75} - x_{25}), x_{75} + 1{,}5(x_{75} - x_{25})], \quad (2.1)$$

где x_{25} – 25%-ная персентиль исследуемого набора чисел;

x_{75} – 75%-ная персентиль исследуемого набора чисел.

Но в нашем случае нет проблем с левым хвостом, потому, предлагаю несколько модифицировать правило – выбросами считаются наблюдения лежащие вне интервала:

$$[0, x_{75} + 1{,}5(x_{75} - x_{25})]. \quad (2.2)$$

Для быстрой реализации этой формулы можно воспользоваться функциями «get_selcond_emiss_25_75» и «cut_emissioins», представленными на листинге В.3. Ключевой в данном случае является «get_selcond_emiss_25_75» она для

переданного столбца типа «pandas.Series» сформирует бинарное условие выбора (массив бинарных значений, совпадающий, по размерности, с исходным – выбираются те значения исходного массива, для которых соответствующие из бинарного условия имеют значение «Истинна») значений не принадлежащих выбросам. Можно так же указать аргумент «constant» (по умолчанию 1.5) – множитель их формулы (2.1) и аргумент «cut_type» (по умолчанию «both») который покажет какую из сторон проранжированного ряда следует обрубить: «both» – обе; «right» – правую; «left» – левую.

Функция «cut_emissioins» надстройка над «get_selcond_emiss_25_75» и прямая реализация (2.2); получает «pandas.Series», возвращает «pandas. Series» без выбросов.

После применения созданных инструментов к столбцам вызвавшим вопросы таблица 2.1 трансформировалась в таблицу 2.2.

Таблица 2.2 – Описательные статистики после очистки от выбросов.

Статистика	Количество фактов просрочки по основному долгу	Максимальное количество дней просрочки	Общее количество запросов в КБ	Сумма кредитных лимитов	Сумма договора	Ежедневный платеж
Число	67783,00	62987,00	100253,00	83282,00	216364,00	224167,00
Среднее	2,04	3,04	7,42	4684,76	2667,00	2,25
СКО	2,89	4,18	5,28	4643,93	2931,37	1,48
Мин.	0,00	0,00	0,00	0,00	0,00	0,00
25%	0,00	0,00	3,00	1000,00	570,80	1,12
50%	1,00	1,00	6,00	3200,00	1118,88	1,76
75%	3,00	8,00	11,00	7000,00	4000,00	3,01
Макс.	12,00	20,00	23,00	20464,32	11600,00	6,67

Примечание – Источник: собственная разработка.

Дальнейшим шагом могла стать борьба с пропусками, но это лучше делать после того, как получены некоторые характеристики классифицирующей способности показателей, потому, как борьба с пропусками обычно связана с необходимостью исключать некоторые столбцы из выборки, а это лучше делать, учитывая взаимосвязи показателей – некоторый показатель может иметь такие сильные классифицирующие свойства, что лучше потерять часть наблюдений, но оставить этот показатель в модели.

По итогам данного раздела мы перешли от первоначальных неочищенных данных к почти полностью обработанным данным. Используя ряд написанных функций отвечающих за процессинг данных удалось: создать столбец отклика, принять решение о удалении номинативных показателей с очень большим числом уровней не подлежащих обработке быстро, удалению всех контрактов относящихся к категории «овердрафт», создание новых показателей, с целью организации выбора на этапе исследования взаимосвязей, преобразование оставшихся показателей типа «Дата» к числовому типу данных и очистка от

невозможных значений и выбросов. Последняя перед процедурой отбора показателей структура данных представлена в таблице Д.3

2.3 Математические методы отбора показателей

ROC анализ является распространенным методом оценки классифицирующей способности моделей. Однако, как будет показано далее, эту методологию можно применить не только как способ оценки качества уже готовой модели, но и как некоторый критерий оценки классифицирующей способности отдельного предиктора.

Рассмотрим общую идею ROC анализа. Пусть у нас получена модель, которая позволяет производить бинарную классификацию. Исследуемый признак Y может относиться к категории у которой не проявилось некоторого признака (Negative – N), или к той, у которой признак проявился (Positive - P). В отношении кредитного скоринга соответственно – клиент без дефолта и с дефолтом.

Тогда интуитивным способом оценить ее производительность, является получение предсказаний на основе модели и построение таблицы следующего вида 2.3.

Таблица 2.3 – Качество бинарного классификатора

Истинный класс	Предсказанный класс		Всего
	N	P	
N	TN	FP	N'
P	FN	TP	P'
Всего	$N*$	$P*$	n

Примечания:

1. Источник: [3 с. 165],
2. TN – число наблюдений для которых модель предсказала отсутствие признака и оказалась права,
3. FN – число наблюдений для которых модель предсказала отсудившие признака, хотя на самом деле он присутствовал,
4. FP – число наблюдений для которых модель предсказала присутствие признака, хотя на самом деле он отсутствовал,
5. TP – число наблюдений для которых модель предсказала наличие признака и оказалась права,
6. $N^* = TN + FN$ – число наблюдений для которых модель предсказала отсутствие признака,
7. $P^* = FP + TP$ – число наблюдений для которых модель предсказала наличие признака;
8. $N' = TN + FP$ – число наблюдений без проявления признака,
9. $P' = FN + TP$ – число наблюдений в которых было замечено наличие признака,
10. $n = N* + P* = N' + P'$ – общее число наблюдений в выборке.

На основании абсолютных чисел сложно делать выводы, потому вводят относительные показатели:

$$FPR = FP/N'$$
$$TPR = TP/P'$$