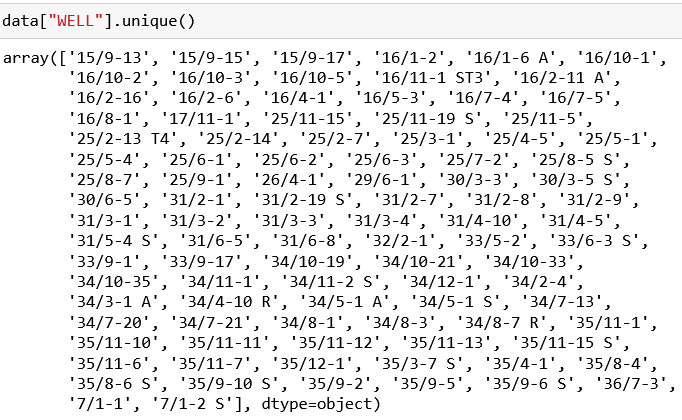
# 1. Базовая информация

## 1.1. Постановка задачи

Имеются 98 скважин, для которых предоставлены:

1. каротажные данные
2. метаданные
3. информация о состоянии скважины
4. полученные литотипы пород по результатам замеров на разных глубинах
5. степень уверенности в определении этих литотипов





Необходимо на основе вышеуказанных данных разработать модель машинного обучения, которая будет определять литотип породы. Оценка точности прогнозов модели проводиться с использованием предоставленной метрики (см. 3.2).

**К метаданным относятся:**

1. **“WELL”:** название скважины
2. **“DEPTH\_MD”:** измеряемая глубина скважины (длина)
3. **“X\_LOC”:** UTM X координата
4. **“Y\_LOC”:** UTM Y координата
5. **“Z\_LOC”:** глубина образца для замера
6. **“GROUP”:** литостратиграфическая группа
7. **“FORMATION”:** литостратиграфическая формация

**К каротажным данным относятся:**

1. **“RDEP”:** Каротаж удельного сопротивления - это метод каротажа скважин основан на определении характеристик породы или отложений путём измерения их электрического сопротивления. Удельное сопротивление показывает, насколько сильно материал сопротивляется протеканию электрического тока. В большинстве случаев электричество переносится ионами в растворе в поровой воде или в воде, заполняющей трещины твердых пород. Если поры породы не насыщены водой, а содержат газы, то проводимость падает, а удельное сопротивление увеличивается. Измерения удельного сопротивления также позволяют различать глинистые и песчаные водоносные горизонты. Также важно понимать, что обычно во время бурения скважины буровые растворы проникают в пласт, изменяя его удельное сопротивление в зоне проникновения.
2. **“RHOB”:** Плотностный каротаж - обеспечивает непрерывную запись объемной плотности пласта по длине скважины. Объемная плотность является функцией плотности минералов, образующих породу, и жидкости, заключенной в поровых пространствах. Это один из трех инструментов каротажа скважин, которые обычно используются для расчета пористости, два других - звуковой каротаж и нейтронный каротаж пористости.
3. **“GR”:** Гамма-каротаж - это метод измерения естественного гамма-излучения для характеристики породы в скважине. Гамма-каротаж позволяет различать сланцы и не сланцы, но не может отличить карбонаты от песчаника. Гамма-лучи ослабевают в зависимости от диаметра скважины, главным образом из-за свойств жидкости, заполняющей скважину. Гамма-каротаж работает через стальные и цементные стенки обсаженных скважин.
4. **“SGR”:** Спектральный гамма каротаж - это метод измерения спектра, т.е. количества и энергии, гамма-лучей, испускаемых естественной радиоактивностью горных пород, связанной с наличием в них калия, тория и урана. Каждый из этих радиоактивных изотопов генерирует гамма-лучи, которые имеют характерный уровень энергии, измеряемый в МэВ. Запись спектроскопической реакции на естественное гамма-излучение обычно отображает массовую долю этих изотопов.
5. **“NPHI”:** Определение нейтронной пористости - это метод, при котором используется источник нейтронов для измерения водородного индекса в резервуаре, который напрямую связан с водонасыщенностью породы. Водородный индекс определяется как отношение концентрации атомов водорода к концентрации чистой воды в одном см3 при 75 °F. Поскольку атомы водорода присутствуют как в воде, так и в нефтеналивных пластах, измерение их количества позволяет оценить величину заполненной жидкостью пористости. К особенностям метода относятся следующие факты: во первых, нейтроны проникают значительно дальше, чем гамма-лучи; во-вторых, для проведения измерений не требуется контакт прибора и стенки скважины; в-третьих, критически важно определить заполненность скважины жидкостью.
6. **“PEF”:** Регистрации фотоэлектрического фактора - это метод, который тесно связан с определением плотности породы. Однако, в отличие от остальных способов, он реагирует именно на минеральный состав матрицы, а не на пористость пласта. Таким образом, он чувствителен к литологии, а не к пористости.
7. **“DTC”:** Акустический каротаж (на модулях сжатия) - это ещё один метод, который может быть использован для расчёта пористости пласта.
8. **“SP”:** Самопроизвольный потенциал (самопотенциал) - это метод, работающий пассивно путём измерения небольших электрических потенциалов (в милливольтах) между глубинами в скважине и заземленным электродом на поверхности. Он помогает в определении глубины проницаемых пластов и их корреляции при сравнении с данными других скважин-аналогов, а также в определении значений удельного сопротивления пластовых вод. На него влияет: толщина пласта, удельные сопротивления в его подошве и прилегающих пластах, удельное сопротивление и состав бурового раствора, диаметр ствола скважины и глубина проникновения бурового раствора в пласт.
9. **“RXO”:** Определение удельного сопротивления промытой зоны - это вспомогательный метод, который используется для расчета насыщения промытой зоны водой, нефтью или буровым раствором, а также для корректировки инструментов глубокого считывания удельного сопротивления (RDEP) с учётом нежелательного проникновения.
10. **“RMIC”:** Резистивиметрия - это комплекс измерений, который проводится с целью определения сопротивления скважинной жидкости. Её выполняют градиент-зондом столь малой длины, что влиянием стенок скважины можно пренебречь.

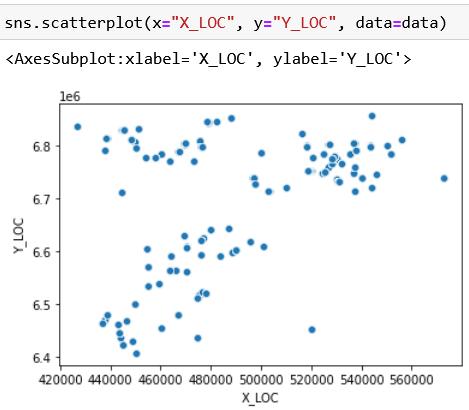
**К данным о состоянии скважин относятся:**

1. **“CALI”:** Метод непрерывного измерения размера и формы скважины по её глубине. Данные значения могут быть важным индикатором размывов, обвалов или набухания сланца в скважине, что может повлиять на результаты большинства каротажных записей.
2. **“BS”:** Диаметр долота, которым проводилось бурение в скважине.
3. **“DCAL”:** Разность истинного диаметра скважины и диаметра бурового долота.
4. **“ROP”:** Скорость, с которой буровое долото разрушает породу под ним, чтобы углубить скважину (скорость проходки или скорость бурения). Как правило, ROP увеличивается в пластах быстрого бурения, таких как песчаники, и уменьшается в пластах медленного бурения, таких как сланцы. Зоны с избыточным или пониженным давлением могут сильно влиять на величину скорости.
5. **“DRHO”:** Поправка на основе Caliper log (CALI), связанная с шероховатостью ствола скважины, для измерения плотности.
6. **“MUDWEIGHT”:** Масса (или плотность) бурового раствора - это одно из его наиболее важных свойств, поскольку оно контролирует пластовое давление, а также способствует стабильности ствола скважины. При низком весе, пластовые флюиды (газ, нефть и вода) будут поступать в ствол скважины, что повышает вероятность его обрушения. В свою очередь, высокая масса может привести к разрывам пласта и попаданию бурового раствора в околоскважинное пространство.

**К интерпретационным данным относятся:**

1. **“FORCE\_2020\_LITHOFACIES\_LITHOLOGY”:** название литологического класса, который определён сторонним интерпретатором.
2. **“FORCE\_2020\_LITHOFACIES\_CONFIDENCE”:** точность определения литологического класса (1-высокая, 2-средняя, 3-низкая).

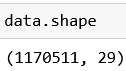
Скважины распределены на плоскости не равномерно. Исходные данные содержат информацию только о координатах скважин в формате UTM без указания номера зоны. В связи с этим, определить точное местоположение области работ невозможно.



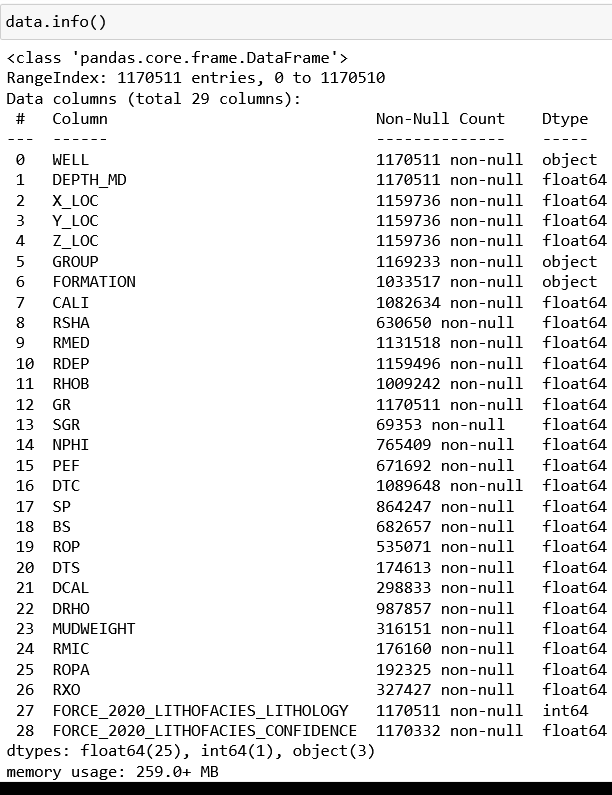
## 1.2. Анализ данных

### 1.2.1. Общая информация

Данные представлены в виде таблицы в формате “.csv” размером **(1170511, 29)** (датасет).



В датасете присутствуют как численные данные, так и категориальные.

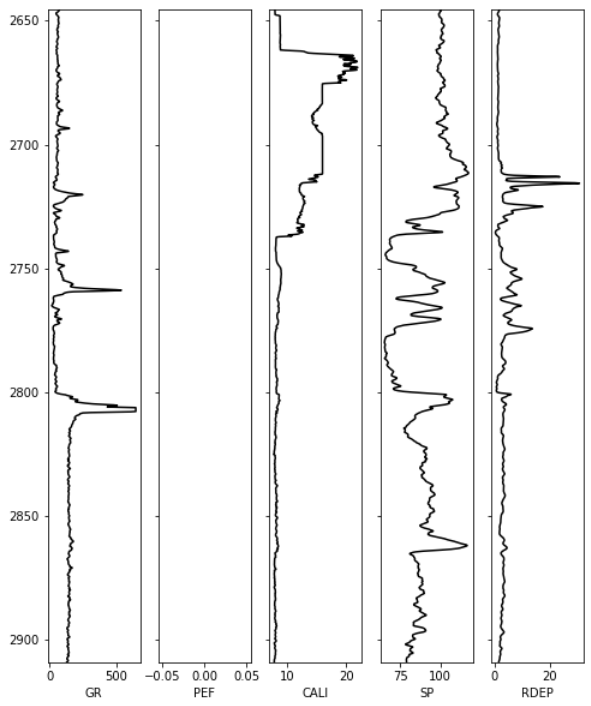
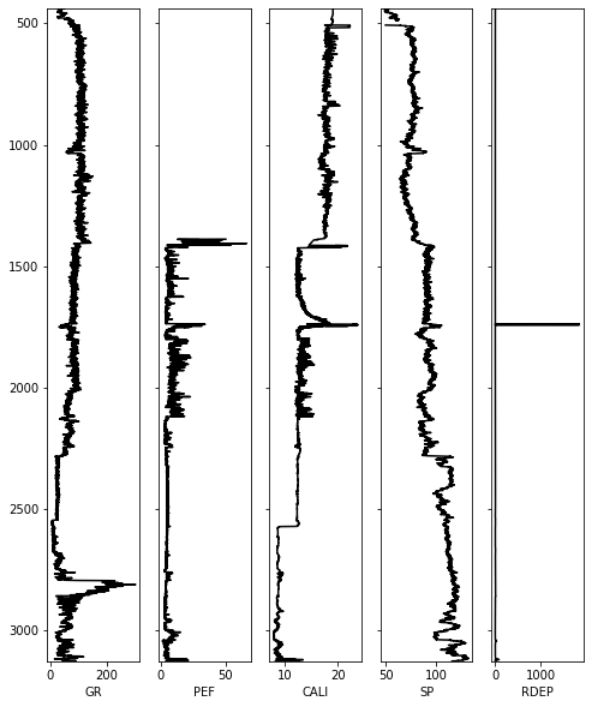


### 1.2.2. Полнота

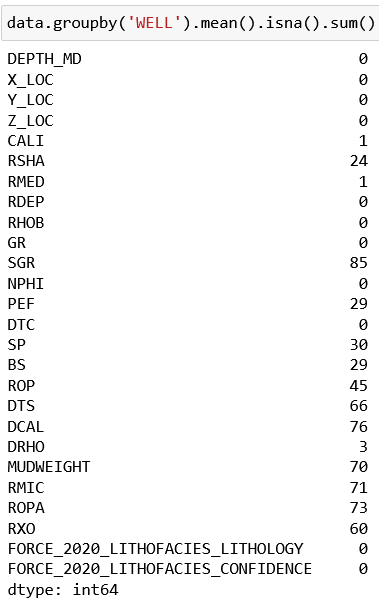
В условии задачи сказано, что в колонках “WELL”, “DEPTH\_MD” и “GR” **гарантированно отсутствуют пропуски**. Если обратить внимание на другие признаки, то можно заметить частичное отсутствие в них данных, при этом количество пропусков для разных колонок (признаков) сильно различается. Литотип (таргет) определён для всех строк таблицы.



Если отобразить несколько каротажных кривых, то видно, что в рамках одной скважины могут отсутствовать как части кривых, так и они целиком.

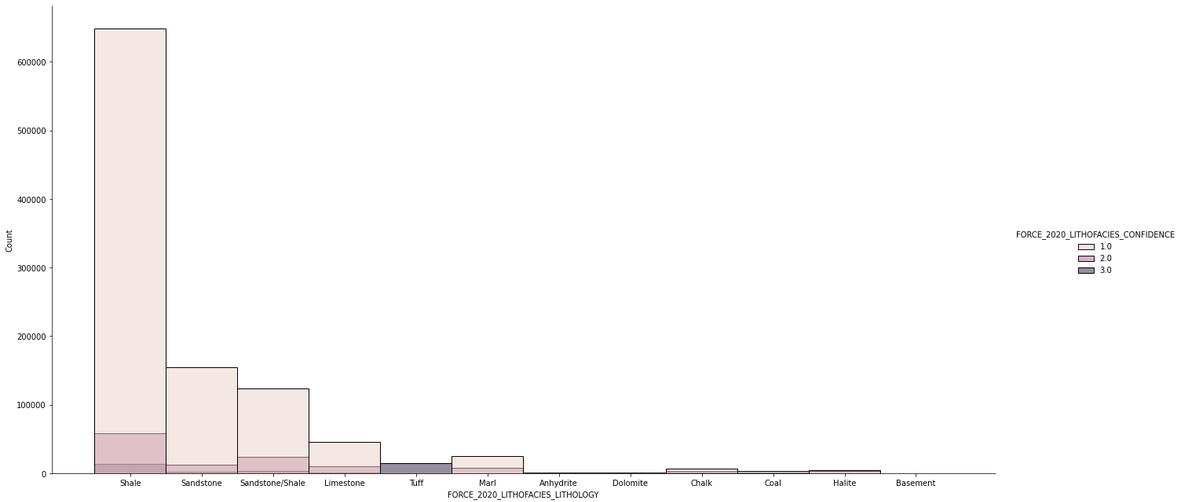


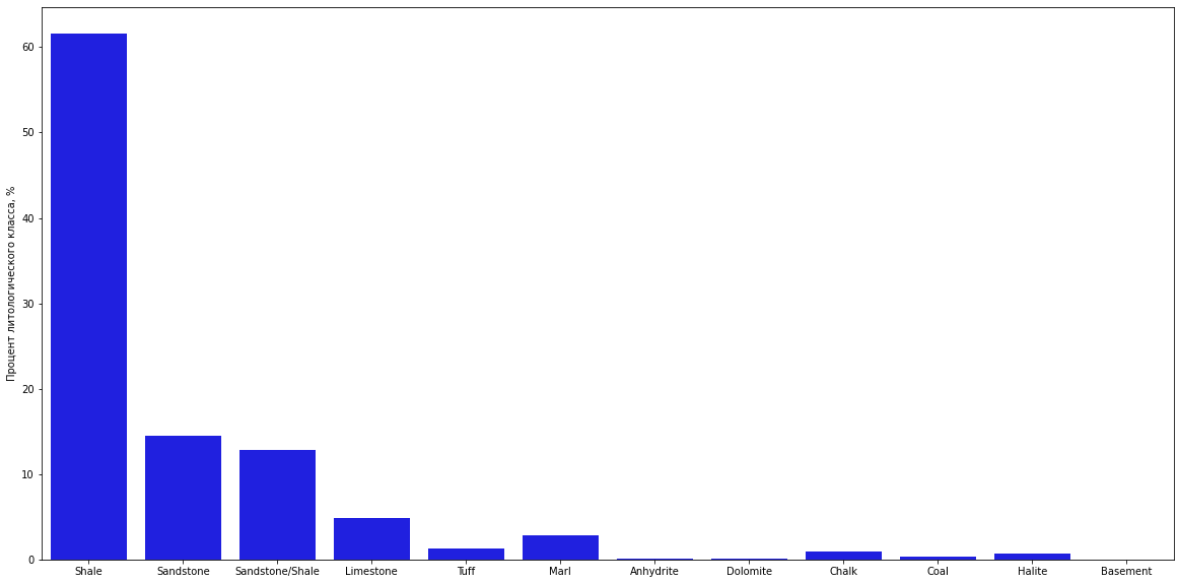
Посмотрим на признаки и у скольких скважин они полностью отсутствуют.



### 1.2.3. Дисбаланс

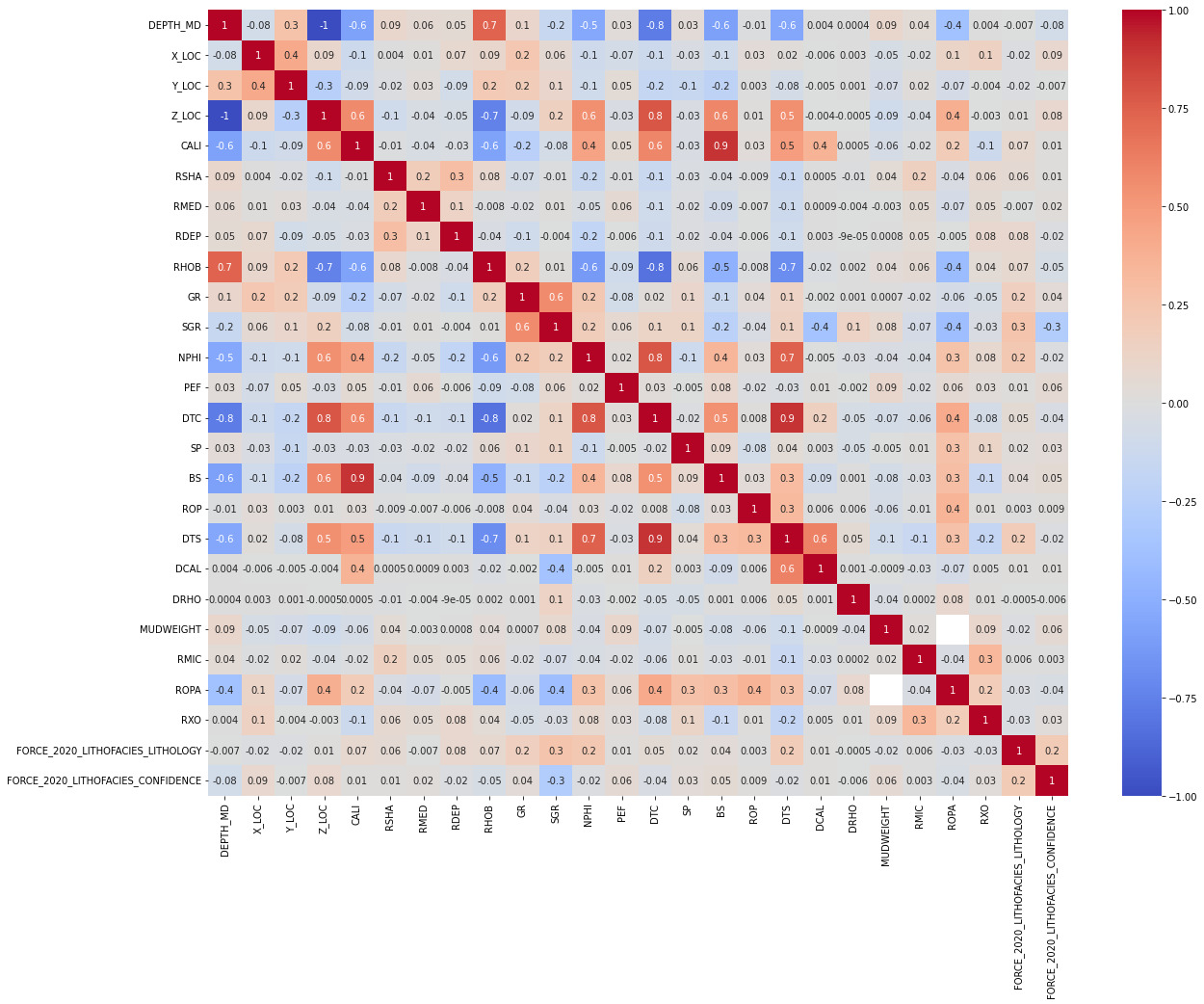
В датасете присутствует **сильная диспропорция литологических классов**, при этом точность определения некоторых литотипов в целом очень низкая.





### 1.2.4. Корреляция данных

Для определения линейной зависимости между признаками используется матрица Пирсона на основе одноименного коэффициента корреляции. В дальнейшем для решения задачи планируется использовать методы градиентного бустинга и бэггинга, которые не чувствительны к мультиколлинеарности, однако она увеличивает время обучения модели и в некоторых случаях приводит к некорректному выделению наиболее значимых признаков. В связи с этим, имеет смысл всё-таки не использовать линейно зависимые признаки.



# 2. Обработка данных

## 2.1 Выбор признаков

В данной задаче признаками могут выступать три категории данных: метаданные, каротажные данные и информация о состоянии скважины.

В качестве начальных данных для решения задачи был предложен набор признаков, который включает в себя как часть каротажных данных, так и неполную информацию о состоянии скважины: 'DEPTH\_MD', 'CALI', 'RSHA', 'RMED', 'RDEP', 'RHOB', 'GR', 'NPHI', 'PEF', 'DTC', 'SP', 'BS'.

В рамках текущей работы предлагается использовать **три дополнительных набора данных**: все предоставленные признаки, только каротажные данные и сочетать в разной степени все три категории данных.

Выбор признаков для последней категории проводился на основе следующих критериев:

1. методика проведения работ ГИС (сочетание каких признаков наиболее информативно с точки зрения методологии анализа данных ГИС)
2. матрица корреляции Пирсона (уменьшение признакового пространства за счёт анализа попарной зависимости фичей)
3. количество пропущенных значений для каждого из признаков (выбирать признаки, для которых количество пропусков минимально)
4. оценка важности признака внутренними инструментами используемых алгоритмов (CatBoost и XGBoost)

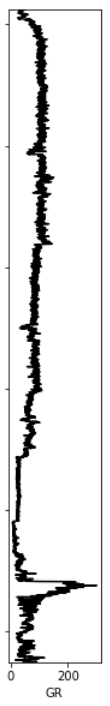
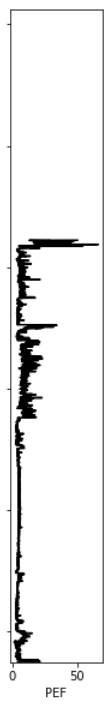
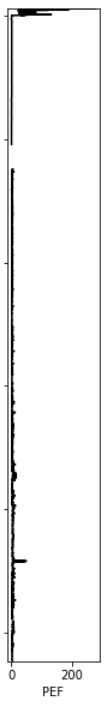
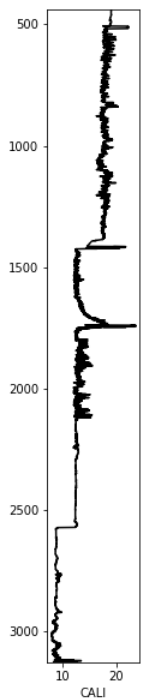
Таким образом, были выбраны следующие признаки: 'WELL', 'FORMATION', 'DEPTH\_MD', 'DRHO','DTC', 'SP', 'GR', 'RDEP',

Стоит отметить, что для корректной предобработки и восстановления данных ко всем наборам признаков необходимо добавить информацию о названии скважин, которая в дальнейшем не обязательно будет учитываться при обучении модели.

## 2.2 Восстановление данных

Для работы планируемых к использованию алгоритмов машинного обучения **обязательно отсутствие пропущенных значений** (NaN), в связи с чем требуется полностью заполнить пробелы в выбранном датасете. Как было отмечено ранее, если отсортировать данные поскважинно, то можно увидеть, что признаки в них могут отсутствовать как частично, так и полностью. Также было показано, что значительное число признаков целиком отсутствует более чем в четверти всех скважин. Нежелательно как совсем отказываться от использования таких признаков, так и значительно сокращать число скважин, на которых модель будет обучаться. Таким образом, подход к восстановлению данных должен учитывать уже имеющиеся измерения в скважине и предоставлять возможность восстанавливать полностью отсутствующие значения.

Восстановление частично имеющихся данных может проводиться за счёт добавления некого усредненного значения в рамках конкретной скважины или путём использования регрессии. При решении данной задачи сфокусируемся на первом способе. Анализ вида каротажных кривых показывает наличие сильных выбросов, однако их природа может быть определена исключительно при комплексном анализе как совместно с другими каротажными кривыми, так и с данными о состоянии скважины. Задача совместной интерпретации подобных наборов признаков является сложным отдельным процессом, и её решение в рамках поставленной цели сильно не соответствует ключевой предметной области, в связи с чем её реализация не планируется. Таким образом, в данных, требующих частичного восстановления, будут присутствовать выбросы, как несущие полезную информацию, так и содержащие нежелательные измерения. Сильный дисбаланс литологических классов также влияет на записи в скважинах для большинства методов ГИС, значения которых не зависят от глубины: на многих каротажных кривых чётко прослеживаются зоны, в которых значения практически не изменяются, а при переходе от одного такого участка к другому, значения резко трансформируются в другой узкий диапазон.



Такие “статические” участки напрямую связаны с фиксированными литотипами или их группами по схожести, а учитывая тот факт, что больше половины всех пород относятся к одному литотипу, можно предположить, что таких зон в рамках каждой отдельной скважины будет немного. В свою очередь, сами значения зашумлены. Устранения шумов также является совершенно отдельной задачей, при решении которой необходимо учитывать особенности каждого метода ГИС, и её реализация не планируется в рамках текущей работы. Учитывая эти особенности, наилучшим значением, которое можно использовать для восстановления такого рода пропусков, является мода, т.к. она не чувствительна к выбросам, и хорошо описывает слабо изменяющиеся данные.

Для восстановления полностью отсутствующих значений предлагаются следующие подходы: рассчитывать средние значение для каждого из признаков по всем литотипам, учитывая данные со всех скважин или принимая во внимания только записи в ближайших скважинах. Второй подход должен быть более точным, т.к. количественные характеристики литотипов могут сильно варьироваться с расстоянием. Для выделения ближайших скважин применяется метод кластеризации K-means, основанный на попарном определении евклидового расстояния между точками и выделении центроидов (центров кластеров).Таким образом, для всех литотипов тем или иным способом вычисляются значения, из которых строятся новые кривые благодаря тому, что литотип (таргет) указан для каждого измерения в датасете.

Однако, крайне важно понимать, что чем больше пропусков заполняется одинаковыми значениями в рамках конкретного литотипа, тем сильнее уменьшается способность модели правильно оценивать новые сторонние данные, несмотря на улучшение метрики. Для предотвращения этой ситуации к восстановленным данным был добавлен Гауссовский шум.

Как упоминалось ранее, в датасете присутствуют категориальные признаки, восстановление которых проводилось с использованием моды.

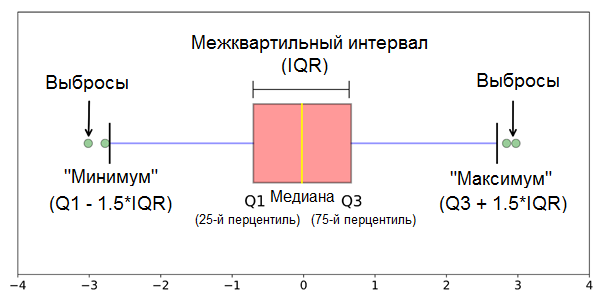
В качестве начального метода восстановления данных в условии задачи предлагалось использовать для заполнения всех пропусков нулевое значение. Такой подход также был протестирован.

## 2.3 Подготовка данных к обучению

### 2.3.1 Скалирование данных

Данная задача предполагает использование признаков, которые сильно различаются как по своей физической природе, так и по величине. Чтобы алгоритмы не отдавали предпочтения тем или иным значениям, которые на фоне других сильно выделяются, например, в силу особенностей того или иного геофизического метода они могут различаться на порядки, имеет смысл масштабировать данные, тем самым их стандартизируя. При этом необходимо обязательно учитывать наличие в них выбросов. Само масштабирование происходит посредством определения центра и диапазона данных.

В текущей задаче использовался Robust Scaler. Принцип его работы заключается в определении медианного значения в качестве нового центра набора данных и интервала, в котором будут отсутствовать выбросы и который будет использован в дальнейшем для масштабирования. Особенность данного метода заключается в использовании межквартильного диапазона (IQR) в качестве интервала для масштабирования. Для вычисления этого диапазона необходимо все данные упорядочить от минимального к максимальному и далее определить квартили. Квартили - это набор отметок на диапазоне распределения вероятностей, которые делят все экземпляры данных на четыре одинаковые части по их количеству. Межквартильным интервалом называется область между первым и третим квартилями по середине которой находиться медиана значений (второй квартиль), а размер этой области составляет половину всей выборки. Таким образом, первый и третий квартили отстоят от медианы на четверть всей выборки в разные стороны. В качестве границ интервала, в который будут попадать данные после скалирования, выбирают значения первого и третьего квартилей минус или плюс соответственно полтора межквартильного интервала.



На основе вышеуказанной информации видно, что **критически важно скалировать данные в рамках именно каждой отдельно скважины**, с целью корректного определения медианных значений и границ интервалов масштабирования. Глубина проведения измерений не масштабировалась.

### 2.3.2 Перемешивание данных

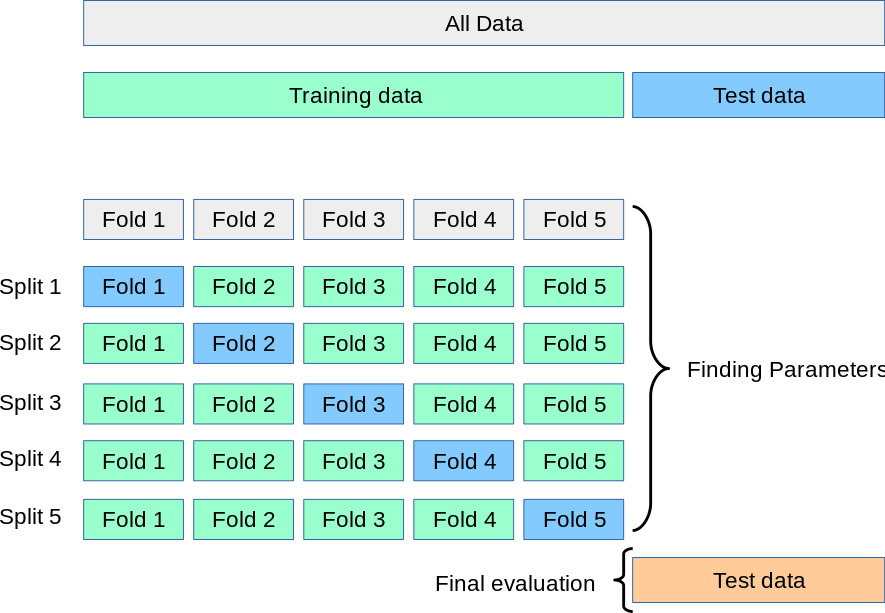
Важным этапом обработки данных является их перетасовывание. Оно проводиться с целью того, чтобы избежать ситуации, когда данные после разбиения на выборки оказываются отсортированными по классам таргета, в частности в тренировочном или тестовом датасетах присутствует таргет только одного типа.

Перемешивание данных обычно производиться при разбиение датасета на тренировочный и тестовый.

### 2.3.3 Разделение на тренировочный и тестовый датасеты

Для эффективного подбора параметров функции прогнозирования в алгоритмах машинного обучения необходимо разделить исходный датасет как минимум на две части: тренировочную и тестовую. Если этого не сделать, то модель будет наиболее точно описывать весь датасет, однако это не значит, что её предсказательная способность будет также хороша на другом наборе данных. Таким образом, произойдёт **переобучение модели**, т.е. её обобщающая способность будет очень низкой и обнаруженные закономерности в тренировочных данных не будет распространяться на новые. Во избежании этой проблемы необходимо использовать как минимум тренировочный и тестовый датасеты: обучать модель на тренировочном и параллельно считать точность на тестовом. Использование тестового датасета позволит предотвратить ситуации, когда модель увеличивает показания на тренировочных данных, однако точность на тестовых уже начинает уменьшаться, т.е. обобщающая способность модели снижается.

Проблема такого подхода заключается в том, что для увеличения обобщающей способности модели следует увеличивать и размер тестового датасета, однако в таком случае есть риск недообучить модель. Альтернативным способом оценки эффективности модели служит метод кросс-валидации, особенность которого заключается в оценке предсказательной способности алгоритма на выборке из тренировочного датасета. Одной из самых распространённых разновидностей такого подхода служит кросс-валидация по k “фолдам”. Принцип такого метода заключается в выделении из тренировочного датасета k выборок и обучении модели на k-1 выборках при дополнительном измерении точности на оставшейся, с последовательным перебором всех выборок в качестве оценочных.

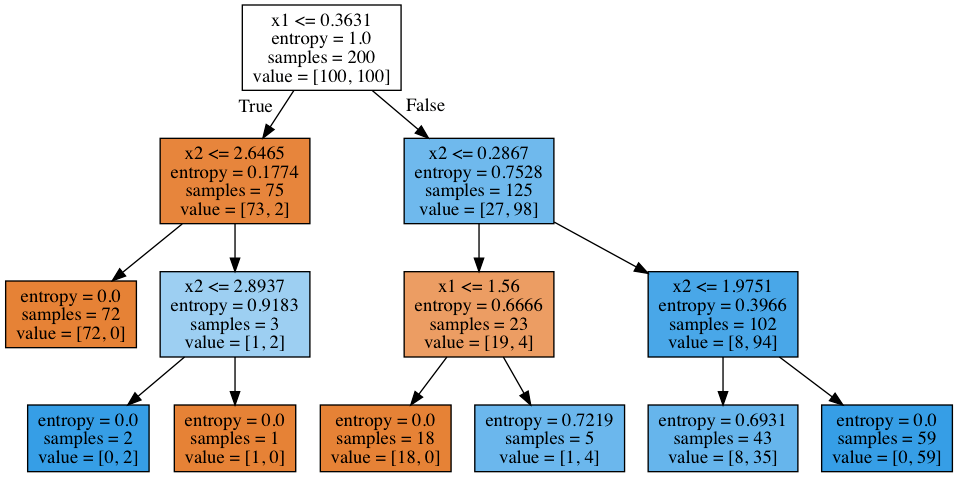


В данной работе использовалось простое разбиение на тренировочный и тестовый датасеты. Процент данных, взятых для обучения, составил 70% от общего размера датасета. Важной особенностью разбиения исходного датасета на выборки при использовании любого из подходов является сохранение изначальной пропорции таргета в созданных подмножествах.

# 3. Обучение моделей

## 3.1 Использованные методы

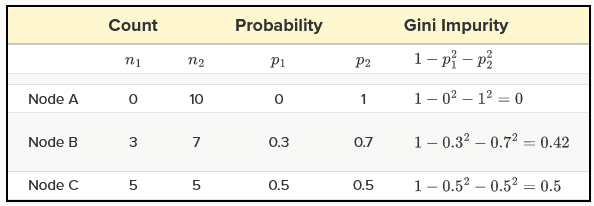
Для решения текущей задачи множественной классификации были выбраны следующие ансамблевые методы: бэггинг и бустинг. В основе этих алгоритмов лежит использование решающих деревьев, которые представляют собой набор простых правил принятия решений, объединённых в древовидную иерархическую структуру. Основными элементами дерева решений являются узлы, которые содержат само решающее правило и на его основе проверяют поступающие в них значения, и листья, содержащие подмножества объектов, которые удовлетворяют всем правилам ветви, заканчивающейся этим листом. Пути, по которым “проходят” значения от первого (корневого) узла до листьев, как раз и называются ветвями. Таким образом, набор значений “пропускается” через решающее дерево и на каждом из его узлов разделяется на всё более мелкие подмножества, а в листьях содержится итоговое разделение.



В задачах классификации объекты, принадлежащие к разным классам, в идеале должны попадать в разные листья, но на практике такое распределение не всегда достижимо из-за необходимости сохранения обобщающей способности модели и/или из-за способа выбора листьев.

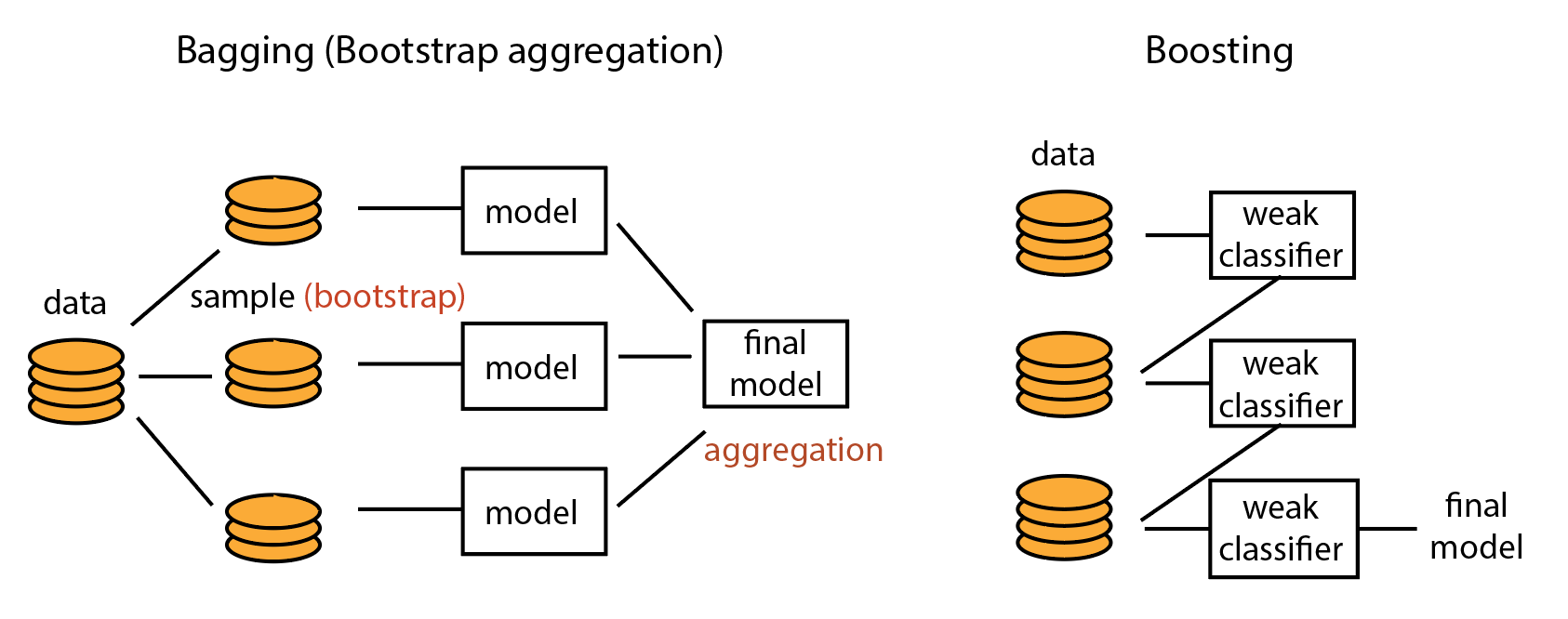
Для лучшего понимания работы данного метода стоит сказать о главных особенностях построения решающих деревьев. Во-первых, необходимо определить условие, когда разделение подмножества больше не требуется, т.е. нужно создать лист. В используемых алгоритмах в качестве такого условия выступает параметр глубины дерева, т.е. максимальное количество узлов в ветви, или минимальное количество выборок, на которые делиться подмножество в узле (по умолчанию 1), т.е. после применения решающего правила подмножество не разделяется.

Во-вторых, стоит описать принцип создания условия в узле. Один из подходов предлагает, чтобы в стартовом множестве содержалась часть объектов из всех классов, обладающих определённым набором признаков. Алгоритм перебирает все эти признаки и на основании их значений для разных классов устанавливает пороговые условия, которые будут использоваться для разделения данных на подмножества. Для оценки качества этих разделений можно использовать, например, коэффициент Джини (меру неоднородности), величина которого характеризует поклассовое распределение элементов: чем ближе значение к нулю, тем больше объектов в выборке принадлежат только к одному классу. В итоге, выбирается разделяющее правило, которое даёт наименьший коэффициент Джини.



Другой подход заключается в том, что на каждом уровне, состоящем из всех последних узлов ветвей равной длины, рассматривается одинаковый признак, а пороговое значение для всех таких узлов устанавливается либо различным, либо одинаковым. Важной особенностью выбранных алгоритмов является то, что **изменение условия узла с целью поиска наилучшего разделения не производится**.

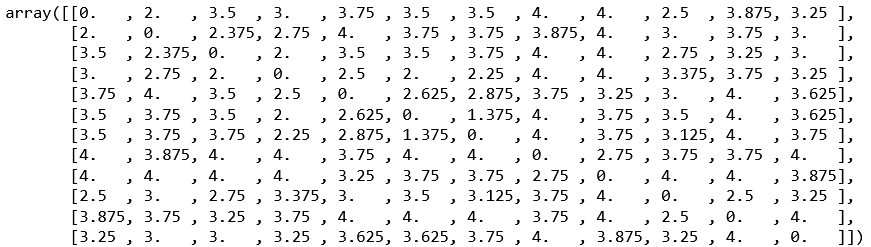
Теперь можно подробно рассмотреть использованные методы. Как упоминалось ранее, они относятся к категории ансамблевых. Это означает, что принцип их работы основан на совместном применении нескольких относительно простых моделей с целью усиления их предсказательных способностей. Ключевая разница заключается в том, что при бэггинге модели используются параллельно и их результаты либо усредняются, либо выбирается наиболее частый, а в случае бустинга, модели обучаются последовательно, при этом каждая новая модель корректирует ошибки предыдущей. Другим отличием является то, что в случае бэггинга исходные данные делятся на выборки (бутсрэпы) для применения в качестве начальных данных для слабых алгоритмов, когда при бустинге используется весь выбранный датасет. Важной особенностью этих подходов является принадлежность всех “малых” моделей к одному семейству, в частности к деревьям решений как в случае выбранных алгоритмов для реализации вышеупомянутых методов.



Для бэггинга был использован алгоритм случайного леса (Random Forest), реализация которого проводилась с применением библиотеки scikit-learn. В случае бустинга были протестированы два программных способа его реализации: с применением библиотек CatBoost и XGBoost. Использование именно двух библиотек для бустинга обусловлено, во-первых, целью изучения особенностей работы с каждой из них, а, во-вторых, выбором метрик для обнаружения переобучения модели из наборов уже реализованных разработчиками, которые частично различаются в этих двух реализация. Здесь подразумевается, что на этапе выбора способа реализации алгоритма бустинга, информация о работе с метриками ещё не была изучена, поэтому хотелось заранее иметь возможность использовать максимальное их количество.

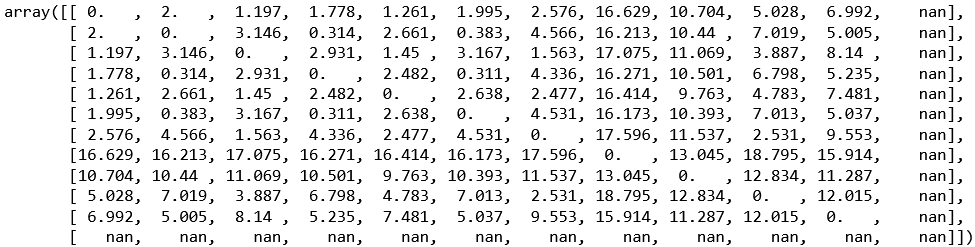
## 3.2 Метрика для оценки финальной точности

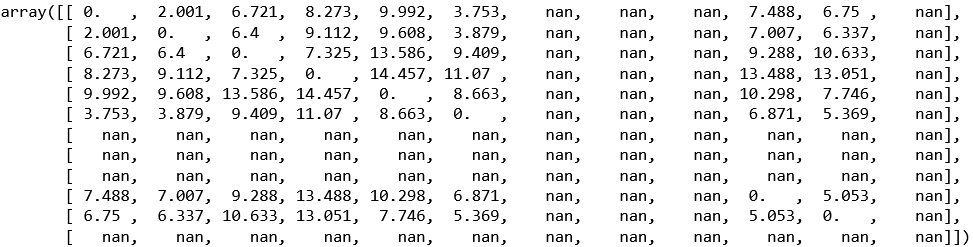
Важно особенностью данной задачи является оценка точности разрабатываемых моделей с применением адаптивной метрики, вид которой зафиксирован в её условии. Использование такого способа оценки позволяет учитывать именно характер связи между объектами, относящимися к разным определяемым классам. Основным элементом данной адаптивной метрики является квадратная матрица штрафов, размером N✕N, где N - количество классов. По одной её оси располагаются истинные значения классов, а по другой - предсказанные. Таким образом, **каждый элемент матрицы находиться на пересечении прогноза и истинного значения** и отражает степень важности разделения данных двух конкретных классов: чем больше записанное значение, тем больше вклад в суммарный штраф за неправильную идентификацию таргета (значения на главной диагонали равны нулю, т.е. класс предсказан верно).

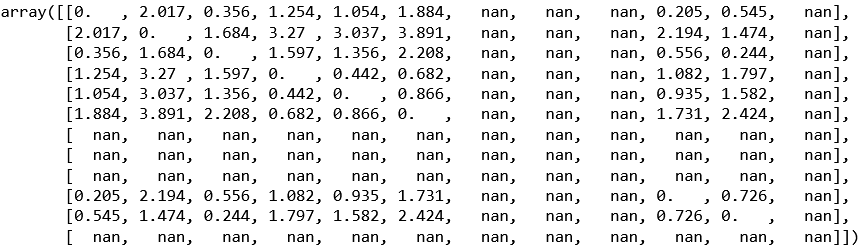


Один из способов построения такой матрицы может основываться на определении евклидового расстояния в m-мерном пространстве для каждой пары классов, где m - количество признаков, для каждого из которых определяется, например, среднее значение и далее на его основе вычисляется расстояние. Применительно к данной задаче, чем более схожи (ближе) литологические классы, тем меньшее евклидово расстояние между ними из-за схожести их физических свойств и как следствие средних значений методов.

Имеет смысл попытаться воспроизвести заданную в условии матрицу штрафов с целью определения примерного набора признаков, на основе которого проводилось её построение, с целью дальнейшего использования этих признаков в модели, исходя из предположения, что способ построения предоставленной матрицы схож с вышеописанным. Рассчитанные собственные матрицы штрафов на основе евклидового расстояния с применением разного количества признаков значимо отличались от матрицы штрафов, заданной в условии задачи. При этом, в исходном датасете присутствуют значительные пропуски, из-за чего для некоторых литотипов полностью отсутствуют записи для ряда методов, что в свою очередь делает невозможным рассчитать для них среднее значение и как следствие определить расстояние в пространстве между всеми выбранными признаками. Таким образом, в рассчитанных матрицах присутствуют пропуски, а остальные элементы хоть и не сильно отличаются от значений в заданной матрице, но их расположение совершенно иное; т.е. выделить искомый набор признаков не удалось. Возможно, что предоставленная матрица штрафов рассчитывалась на ином наборе данных или в основе метода её определения лежал другой подход.







### 3.2.1 Веса классов

Дополнительным способом учёта матрицы штрафов при обучении модели служит создание на её основе весовых коэффициентов для каждого из классов. Использовался следующий подход: для каждого столбца (класса) матрицы было определено среднее значение, величина которого характеризует общую степень важности разделимости выбранного класса и остальных. Чем больше это значение, тем важнее обучить модель правильно определять данный класс.

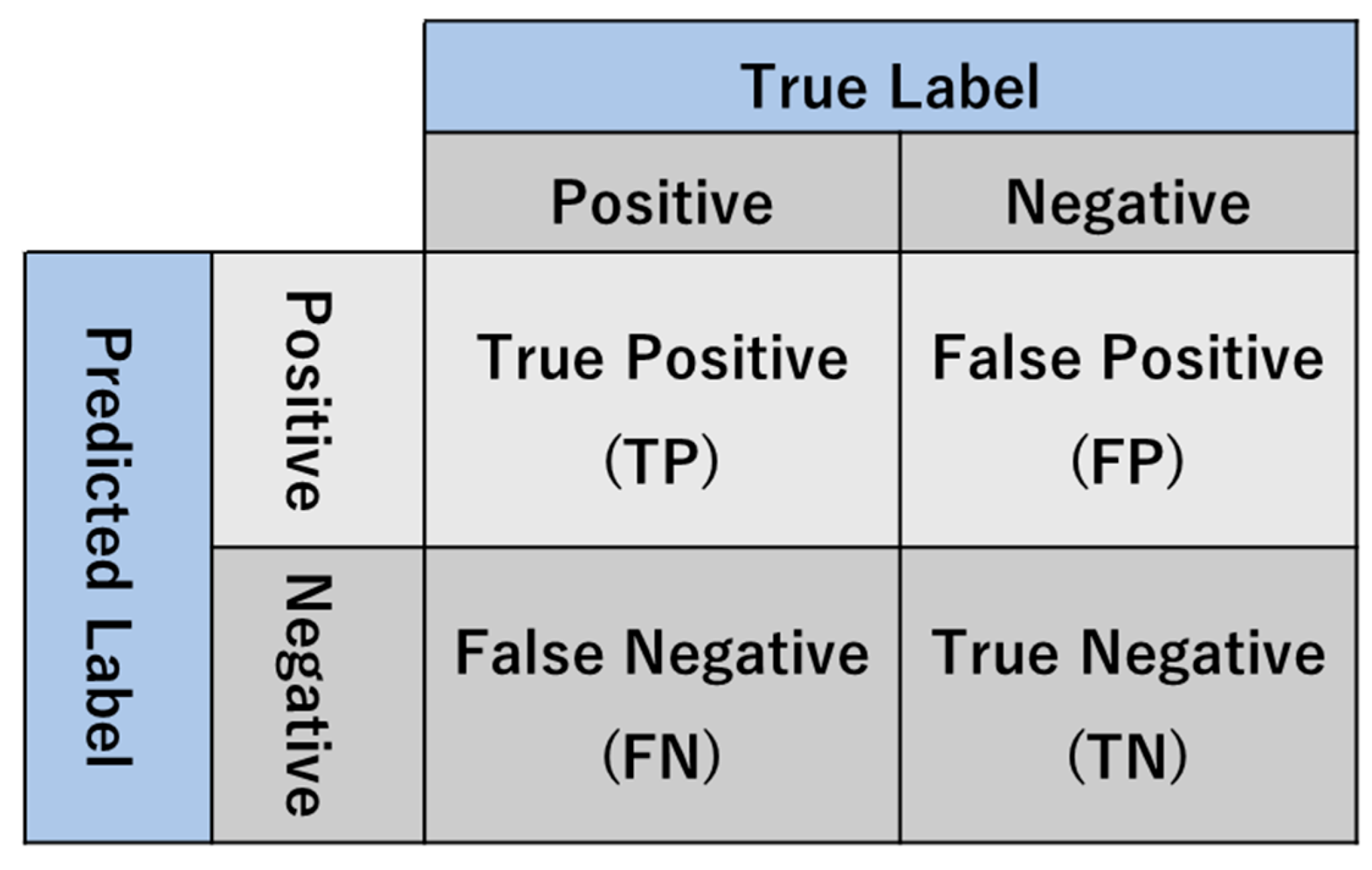
## 3.3 Возможные способы борьбы с дисбалансом классов

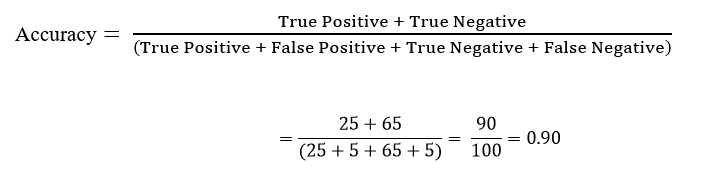
Как упоминалось ранее, в датасете присутствует сильная диспропорция классов, которая может повлиять на их разделимость моделью. При использовании деревьев решений и базирующихся на них алгоритмов, в текущем случае бэггинге и бустинге, дисбаланс классов влияет на меру неоднородности данных в листья (коэффициент Джини или энтропию).

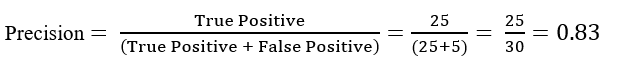
Наиболее распространенными методами для борьбы с несбалансированностью классов являются подходы **недосемплирования** (уменьшение количества элементов больших классов) и **пересемплирования** (увеличение количества элементов малых классов). Также возможно **добавление весовых коэффициентов** объектам классов в зависимости от количества экземпляров: чем меньше в классе объектов, тем больший коэффициент им присваивается.

Однако, наличие дисбаланса классов не всегда требует изменения начальных данных или создания на их основе весовых коэффициентов. Во многом, в такой ситуации выбор стратегии зависит от метрики оценки качества обучения модели.

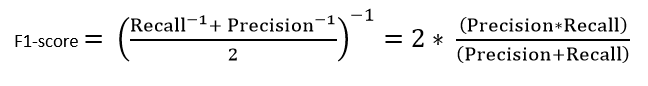
Существуют метрики как не чувствительные к дисбалансу за счёт работы именно с вероятностями относимости объектов к классам, а не с самим фактом их принадлежности к ним, так и сильно зависящие от равномерности распределения экземпляров, работающие с дискретными значениями классов. В первом случае перебалансировка данных приведёт к тому, что вероятность принадлежности объекта к определённому классу будет сильно отличаться от вероятности того, что данный прогноз модели верный, т.е. алгоритм будет неоткалиброванным. Во втором случае требуется применить способы борьбы с несбалансированностью данных, т.к. высокая точность в такой ситуации может достигаться за счёт определения объектов малых классов в большие. В качестве метрик, работающих с вероятностями, можно выделить ROC AUC и LogLoss. Первая хорошо применима в случае бинарной классификации, но не так эффективна как LogLoss при многоклассовой. Альтернативой им выступают метрики Accuracy и F1-мера, работающие с дискретными значениями классов. F1-мера более показательна при дисбалансе классов из-за учёта ошибочной классификации, когда Accuracy берёт во внимание только правильно определенные классы.





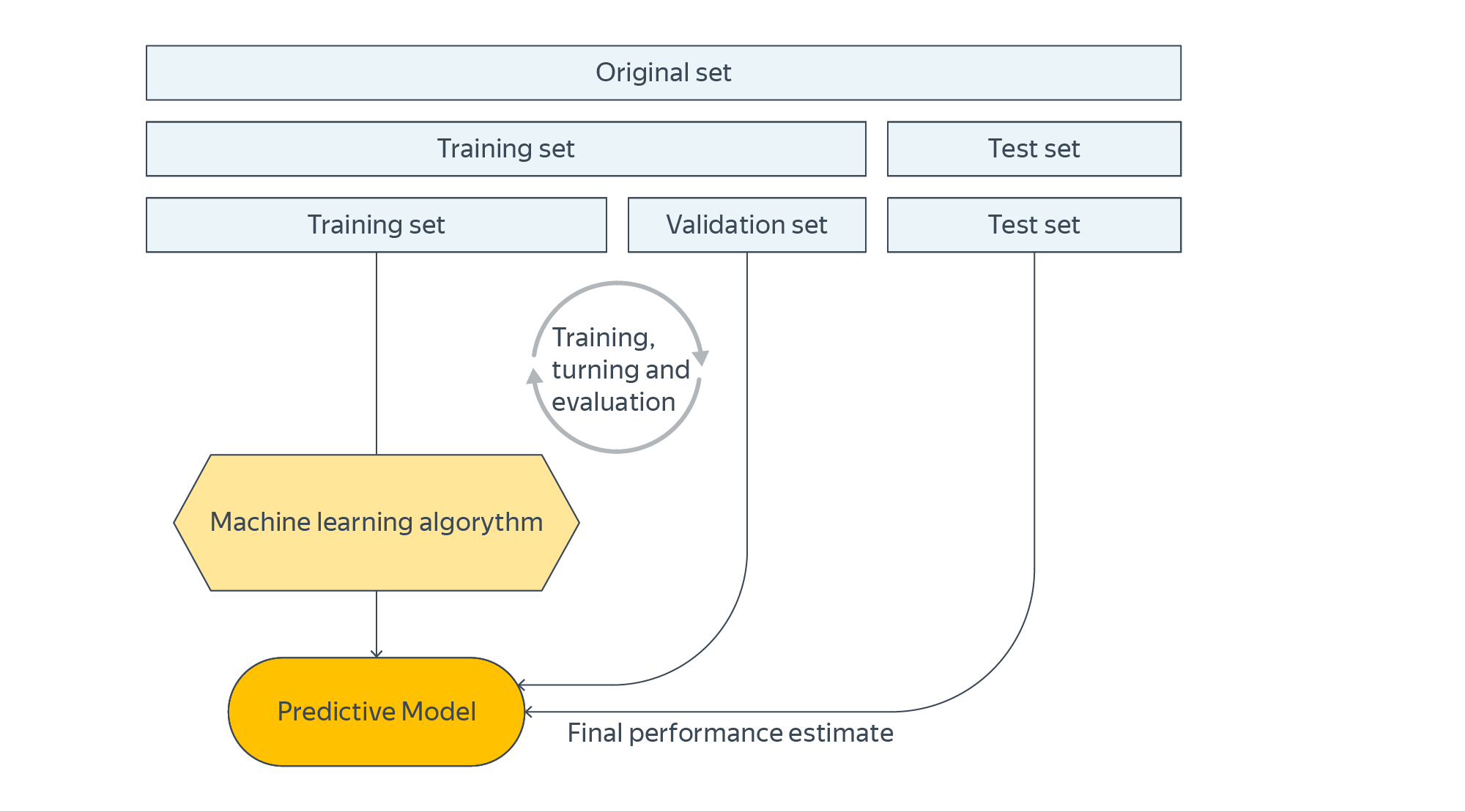






## 3.4 Подбор гиперпараметров моделей

Для правильного проведения обучения модели необходимо настроить её гиперпараметры, которые непосредственно **используются для управления процессом тренировки**. Их ключевым отличием от весов модели или структуры решающего дерева является то, что их значения вычисляется, основываясь не на метриках или элементах датасета, а определяются в ходе итеративного обучения.



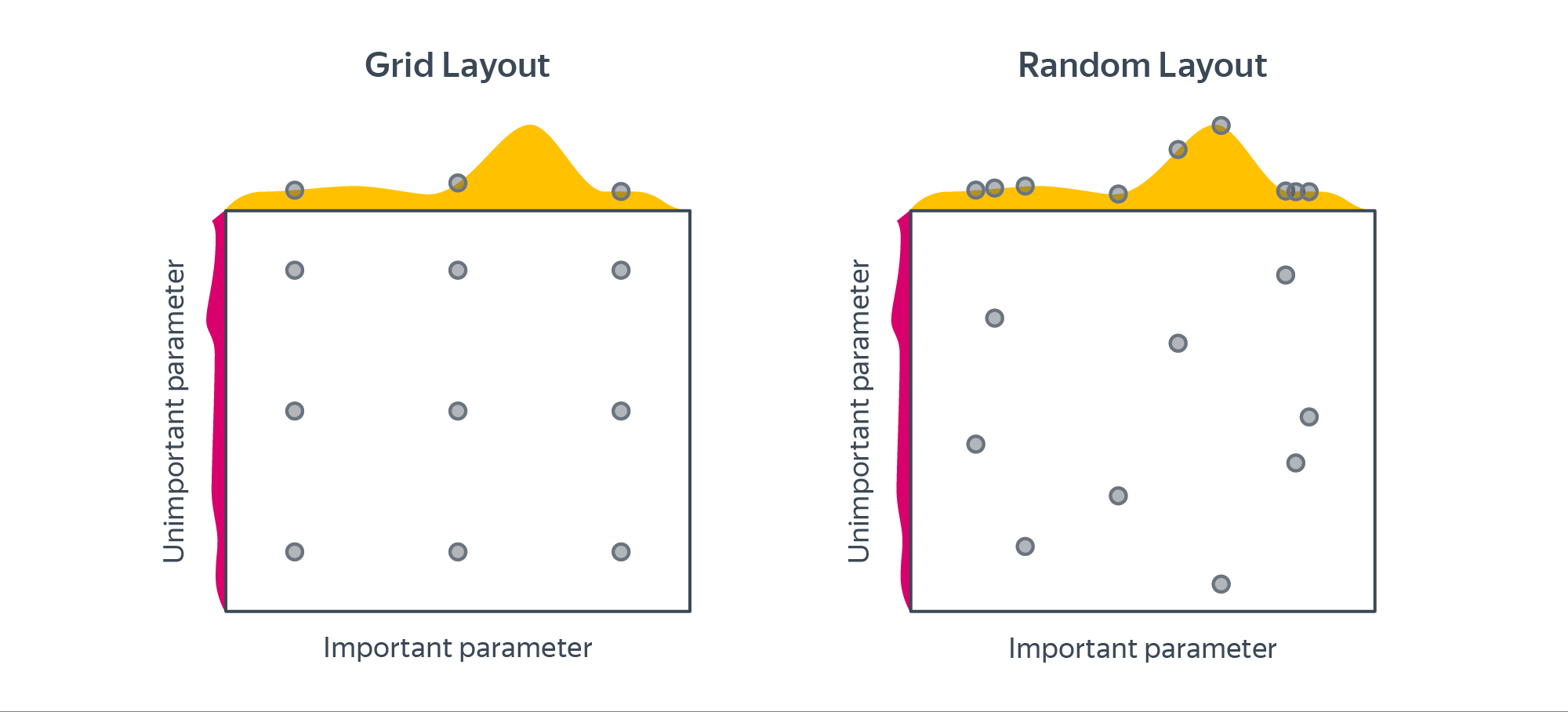
Таким образом, гиперпараметры напрямую не влияют на качество прогнозов модели, а лишь на процесс её построения.

### 3.4.1 Способы подбора гиперпараметров

Существует несколько подходов по поиску гиперпараметров, однако принцип у них один и тот же - перебрать ряд значений в определённых диапазонах, оценивая при этом как скорость, так и качество построения модели по одной из метрик. Такая оценка может проводиться с использованием кросс-валидации или с выделением одной фиксированной части из тренировочных данных.

Рассмотрим несколько основных способов перебора значений гиперпараметров. Во-первых, можно “пройти” по всем значениям в фиксированных диапазонах (или по их логарифмическим представлениям) с заданным шагом, оценив все возможные их комбинации, с целью поиска наилучшей. Такой подход называется **Grid Search**, и для него сразу можно выделить основное ограничение - требуется большое количество времени для тренировки модели с каждым набором значений.

Во-вторых, можно перебрать не все комбинации значений гиперпараметров, а только их случайное подмножество. Данный способ называется **Random Search**. Он позволяет уменьшить количество анализируемых значений, а значит и суммарное время обучения, и при этом оставляет вероятность найти наилучший набор без сокращения диапазона поиска. Более того, одни гиперпараметры могут быть важнее в рамках конкретной задачи, чем остальные, а значит необходимо подобрать оптимальные значения для наибольшего их количества. Random Search как раз позволяет сделать рассматриваемый набор значений более разнообразным в сравнении с Grid Search, за счёт чего повышает вероятность обнаружения оптимальной комбинации за одинаковое количество итераций.



Общий недостаток вышеописанных подходов заключается в отсутствии возможности учитывать результаты прошлых вычислений. Имеет смысл отдельно исследовать как области с небольшим количеством проверенных значений, надеясь обнаружить там более подходящие, так и зоны с большей изученностью, где поиск оптимальных значений более осмысленный. Комбинация данных принципов в разной степени представлена в отдельной группе методов. Например, алгоритм **Tree-structured Parzen Estimator (TPE)**, на каждой новой итерации обучения модели выбирает гиперпараметры как раз на основе прошлых результатов. Первые несколько итераций алгоритм действует по принципу Random Search, выделяя затем группу наиболее удачных значений для каждого из гиперпараметров. Далее вокруг этих значений определяются области, в которых будет производиться поиск наиболее успешных “кандидатов”. Затем происходит проверка одного из кандидатов (значений), и на основании метрики корректируется область дальнейшего поиска. Также алгоритм учитывает влияние одних гиперпараметров на другие.

В данной задаче поиск гиперпараметров проводился с использованием библиотеки Optuna, работающей на основе усовершенствованной версии TPE (Adaptive-TPE).

# 4. Ход работы и результаты

## 4.1 Эксперимент №1

В качестве начального способа построения модели в условие задачи предлагался следующий вариант: заполнить все пропуски в данных нулевым значением, провести скалирование сразу для всего датасета с использованием Standard Scaler (стандартного отклонения) и воспользоваться алгоритмом Random Forest для построения прогнозов.

Для реализации первого собственного способа решения задачи был выбран алгоритм CatBoost, способ обработки данных не изменился, а настройка гиперпараметров не проводилась. Такой подход показал ухудшение точности по итоговой метрике по сравнению со стартовым вариантом.

Далее было принято решение использовать алгоритм XGBoost также без настройки гиперпараметров и изменения способа обработки данных. Точность в данном случае также ухудшилась, но значительно на большую величину. Таким образом, можно сделать вывод, что алгоритм CatBoost имеет лучше заданные по умолчанию гиперпараметры.



В данном случае, низкая точность моделей при использования алгоритмов бустинга в большей степени связана с условиями окончания построения деревьев решений и количеством этих деревьев. По умолчанию, данные алгоритмы руководствуются параметром глубины дерева, т.е. максимальной длины ветви, в качестве условия создания листьев, а число последовательно использующихся деревьев невелико. В случае метода Random Forest, по умолчанию, дерево строиться до тех пор, пока в его листьях коэффициент Джини не станет нулевым и при этом количество параллельно использующихся деревьев сопоставимо с числом итераций надстройки таковых при бустинге, т.о. достигается более высокая описательная способность этого алгоритма, когда **модели CatBoost и XGBoost получаются недообученными**.

## 4.2 Эксперимент №2

Таким образом, для корректной работы всех алгоритмов, было принято решение в первую очередь настроить гиперпараметры моделей бустинга с использованием библиотеки optuna на основе метода TPE; а также для предотвращения переобучения моделей во время их тренировки дополнительно считать точность на тестовых данных по метрике Accuracy. Была выбрана наиболее распространённая метрика без “опоры” на теорию оценки точности моделей, учёт дисбаланса классов не проводился. Дополнительно, для каждого класса были рассчитаны коэффициенты на основе матрицы штрафов, путём вычисления среднего значения для каждого столбца матрицы, которые также использовались при обучении моделей.



В результате, точность по адаптивной метрике выросла для обоих алгоритмов, достигнув, а в некоторых случаях превысив, базовые значения метода Random Forest.

### 4.2.1 Работа алгоритма XGBoost с видеопамятью

На данном этапе была обнаружена проблема в использовании библиотеки XGBoost, связанная с её некорректной работой с видеопамятью графического ускорителя: после каждого обучения модели, видеопамять очищалась не полностью. Это приводило к постепенному уменьшению её доступного объёма для тренировки новых моделей, в частности, CatBoost для корректной работы требовал более 70% всей видеопамяти, которой не хватало после обучения моделей XGBoost. Однако, при подборе гиперпараметров для алгоритма XGBoost такой проблемы не возникало. Причина заключалась в том, что данная библиотека тесно интегрирована с другой, упоминавшейся ранее - scikit-learn. Это даёт возможность пользователю применять многие функции из сторонней библиотеки при настройке моделей XGBoost, в частности использовать при тренировке данные не в “родном” формате библиотеки - DMatrix, а в виде объекта Pandas DataFrame. Как раз **при использовании стороннего формата данных, библиотека практический не очищала видеопамять**, однако при их представлении в виде DMatrix (разреженной матрицы), алгоритм освобождал большую часть занятой видеопамяти, хотя тоже не полностью. Важным изменением стало то, что при тренировке новых моделей, количество занимаемой памяти в таком случае не увеличивалось, а оставалось таким же, как после обучения одной единственной модели. При подборе гиперпараметров для XGBoost как раз использовалось представление данных через DMatrix, а при обучении - в виде Pandas DataFrame, поэтому обозначенная проблема не возникала на первом этапе. Таким образом, для её решения необходимо использовать формат DMatrix. В случае работы с библиотекой CatBoost, объём занимаемой видеопамяти при обучении был в несколько раз больше, чем у XGBoost, однако она полностью очищалась как при использовании формата Pandas DataFrame, так и при представлении данных в “родном” формате библиотеки - Pool.

## 4.3 Эксперимент №3

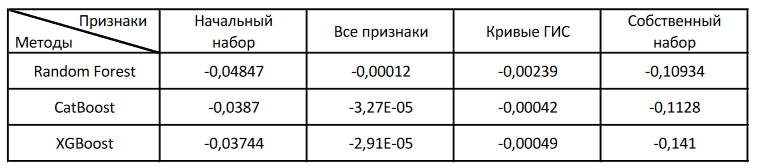
Далее, был подробно изучен материал на тему оценки качества создаваемых моделей, и принято решение протестировать LogLoss для бустинга. В среднем, использование данной метрики при контроле переобучения модели, привело к улучшению финальной точности по адаптивной метрике, поэтому было принято решение дополнительно не тестировать способы оценки, работающие с дискретными значениями классов и требующие дополнительной перебалансировки датасета или введения весовых коэффициентов, такие как Accuracy или F1-мера.



Модель Random Forest была предложена в условии задачи с уже частично выбранными гиперпараметрами. В дальнейшем, их настройка не проводилась, т.к. модель использовалась исключительно для сравнения разрабатываемых собственных алгоритмов на основании бустинга с подходом, предлагаемым в условии задачи.

## 4.4 Эксперимент №4

Таким образом, на этом было принято решение закончить с настройкой моделей и перейти к обработке данных, в частности к способу заполнения отсутствующих значений. Как упоминалось ранее, в датасете отсутствуют как части кривых, так и они целиком. Восстановление не полностью пропущенных элементов в рамках фиксированных скважин проводилось на основе вычисления для части признаков мод значений и заполнения ими пробелов в данных. Заполнение же полностью отсутствующих кривых проводилось следующим образом: для каждого литотипа вычислялись медианные значения по всему датасету для всех признаков, а затем ими же заполнялись кривые, с учётом того, что таргет (литотип) указан для всех записей датасета. Также скалирование данных проводилось поскважинно, а не в рамках всего набора значений, с использованием Robust Scaler, а не Standard Scaler.



Как можно видеть, такой подход значительно увеличил точность работы каждого из алгоритмов, однако **сразу возникают опасения о переобученности моделей**. Для предотвращения эффекта переобучения, во время тренировки моделей бустинга, на тестовых данных вычисляется точность по метрики LogLoss, и если она не улучшается в течении определённого количества итераций, то модель перестаёт обучаться, а в качестве её итогового варианта остаётся конфигурации с наилучшей точностью по выбранной метрике. Такой подход основывается на том, что тестовые данные значимо отличаются от тренировочных, за счёт чего сохраняется высокая обобщающая способность модели. Как упоминалось ранее, значительная часть признаков полностью отсутствует в скважинах, т.е. их восстановление производится медианными значениями, одинаковыми для каждого отдельного литотипа. Более того, практический 60% всех записей в датасете принадлежат к одному литотипу. Таким образом, заполненный датасет состоит из большого числа замеров, в которых значения для ряда признаков абсолютно не изменяются. Это потенциально приводит к тому, что данные в тестовой и тренировочной выборках сильно похожи (как и в рамках каждой из них), а значит контроль переобучения по такому набору данных проводить некорректно, также как и использовать кросс-валидацию.

## 4.5 Эксперимент №5

Будем считать, что заполнение частей кривых модой в рамках отдельных скважин создаёт незначительное количество дублированных значений, а данная проблема потенциально может возникать всё-таки из-за полностью “пустых” признаков. Таким образом, разумно изменить подход к заполнению последних: предлагается вычислять медиану не по всем строкам датасета, а только на основе записей из ближайших скважин. Их группировка проводилась по паре координат (X, Y) методом k-means из библиотеки scikit-learn. Также было принято решение добавить ко всем данным Гауссовский шум величиной 15% от стандартного отклонения для всех признаков в рамках каждой отдельной скважины, чтобы полностью избавиться от одинаковых значений.



В результате, точность по финальной метрике для всех наборов признаков ожидаемо уменьшилась, однако её изменение было неравномерным. Больше всего точность “пострадала” для наборов с большим количеством признаков: для кривых ГИС и для всех признаков датасета; а изменение для группы собственно выбранных признаков оказалось самым незначительным. Это связано с тем, что одним из условий выбора видов данных для последнего набора была их максимальная полнота, т.е. в нём присутствовало минимальное количество восстановленных значений; когда в случае больших наборов признаков, восстанавливалось намного большее количество записей. Таким образом, можно сделать вывод о том, что **модели всё-таки переобучались при заполнении “пробелов” в датасете одинаковыми значениями для каждого из литотипов**. Добавление Гауссовского шума и усреднение значений по группам скважин, вероятно, повысило их обобщающую способность, однако можно дополнительно проанализировать четыре итоговых варианта, чтобы выбрать наиболее устойчивое сочетание признаков. Результат по финальной метрике для собственного набора данных практически не изменялся в зависимости от способа восстановления значений (нулями или модами вместе с медианами), когда для других протестированных сочетаний признаков точность отличалась на порядки.

## 4.6 Итоговая модель

Основываясь на этом факте, в качестве модели с наилучшими обобщающими способностями и с наибольшей точностью по финальной метрике (при сравнении трёх алгоритмов с фиксированным способом предобработки данных) была выбран модель XGBoost, обученная на собственном наборе данных, которые восстанавливались медианными значения для групп скважин по близости их расположения в пространстве и модами в рамках каждой отдельной скважины.