Научный Исследовательский Институт

“Высшая Школа Экономики”

Факультет информатики, математики и компьютерных наук

**Разработка модели на основе алгоритма Случайного леса для задачи прогнозирования успеваемости студентов.**

(курсовая работа)

Выполнил студент

2-го курса гр. 20ПМИ-2

Блох А. М.

Научный руководитель

Шадрина Е. В.

Нижний Новгород 2022

**Оглавление**

[**Введение** 2](#_Toc104211330)

[**Постановка задачи** 3](#_Toc104211331)

[**Методы решения** 4](#_Toc104211332)

[**1) Решающие деревья** 5](#_Toc104211333)

[**2) Бэггинг** 12](#_Toc104211334)

[**4) Метрики для сравнения алгоритмов** 16](#_Toc104211335)

[**5) Предсказывание успеваемости студентов** 18](#_Toc104211336)

[**Вывод** 21](#_Toc104211337)

[**Список литературы** 24](#_Toc104211338)

# **Введение**

В настоящее время машинное обучение является очень популярным направлением в программировании и широко используемым и эффективных инструментом для решения прикладных задач в большинстве бизнесов, отраслей производства и в науке. С помощью большого количества разных алгоритмов, машинное обучение позволяет прогнозировать большое количество разных вещей, распознавать объекты на фотографиях, создавать чат-ботов, почти не отличимых от людей по общению и решать огромное количество других сложных задач.

В машинном обучении есть две популярные общие задачи:

* Регрессия
* Классификация

Задача регрессии подразумевает предсказывание моделью машинного обучения численной метки у объекта. Например, основываясь на различных экономических показателях за конкретный, предсказать инфляцию в стране в процентах в этом году.

Задача классификации заключается в том, чтобы отнести объект к одному из нескольких классов. Например, основываясь на данных о человеке, определить его кредитоспособность и решить, выдать ему кредит или нет, то есть отнести к одному из двух классов: кредитоспособный и некредитоспособный.

В дальнейшем речь пойдет именно о задаче бинарной классификации.

# **Постановка задачи**

Имеется таблица с результатами опроса студентов, содержащая различные вопросы, основываясь на которых, можно как-то предположить успеваемость студента. Также на сайте ВШЭ есть таблицы, в которых можно посмотреть успеваемость студентов, участвовавших в этом опросе.

Задача состоит в том, чтобы с помощью алгоритма случайного леса обучить модель машинного обучения на результатах опроса и данных об успеваемости студентов, чтобы она могла по новым входным данным предсказывать, будут ли у студента неудовлетворительные оценки.

Если говорить более формально перед нами задача бинарной классификации, общая формулировка которой выглядит следующим образом:

Пусть X — множество признаков объектов, Y — множество классов. Существует неизвестная целевая зависимость - отображение , значения которой известны только на объектах обучающей выборки . Требуется построить алгоритм, который будет уметь классифицировать произвольный объект .

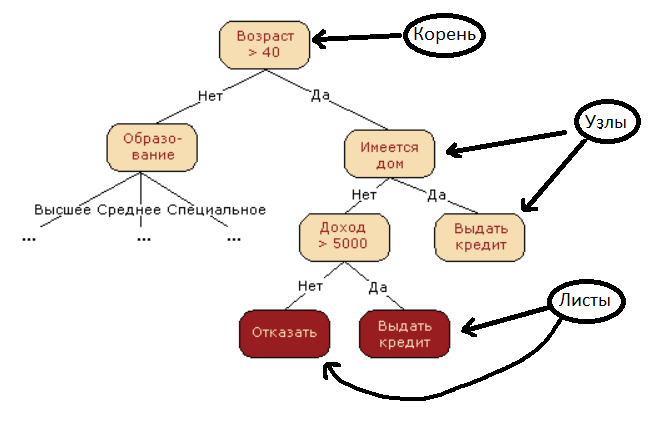
# **Методы решения**

Прежде чем в подробностях описывать, как я решал эту задачу, нужно разобраться что такое алгоритм случайного леса и как он работает, а для этого в свою очередь нужно понимать, как работают решающие деревья (decision trees).

## **1) Решающие деревья**

***1.1) Определение:***

*Решающее дерево (decision tree) - это модель машинного обучения, которая предсказывает значение целевой переменной, применяя последовательности простых решающих правил, которые называются предикатами.*



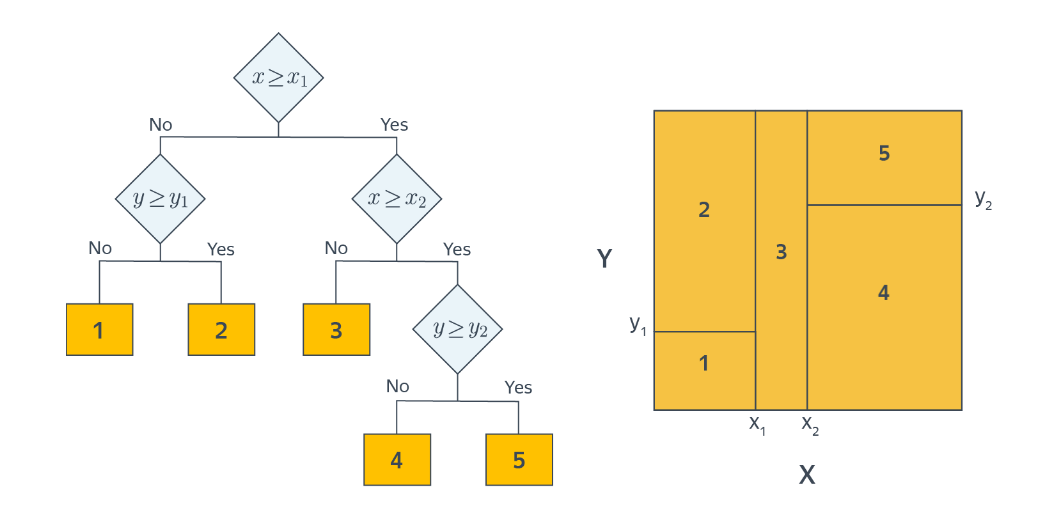
***1.2) Для чего используются решающие деревья:***

Решающие деревья широко используются в машинном обучении в задачах:

* Классификации (процесс группирования объектов по категориям на основе предварительно классифицированного тренировочного набора данных)
* Регрессии (процесс прогнозирования числового значения целевой переменной)

***1.3) Пример:***

Пусть наша задача состоит в том, чтобы определить к какому из 5 классов относится объект. Вот пример решающего дерева для такой задачи:



* Объекты в этом примере имеют два вещественных признака X и Y. Решение о том, к какому классу будет отнесён конкретный объект выборки, будет приниматься путем прохода от корня дерева к одному из его листов.
* В каждом узле этого дерева находится условие (предикат). Если условие верно для текущего объекта из выборки, то мы переходим в правого потомка, иначе — в левого. В этом примере все условия — это просто взятие порога по значению какого-либо признака.
* На картинке справа от дерева мы видим решающую поверхность, построенную данным решающим деревом. Таким образом, каждый объект из выборки будет отнесён к одному из пяти классов в зависимости от того, в какую из областей поверхности он попадает.

***1.4) Как строится решающее дерево:***

Главный вопрос, который напрашивается сам собой - это как искать лучшее разбиение для дерева. Это делается с помощью такой величины, как прирост информации (Information gain, позже обозначается IG). IG можно считать по-разному, например, с помощью Энтропии (Entropy) или с помощью индекса Джини (Gini impurity).

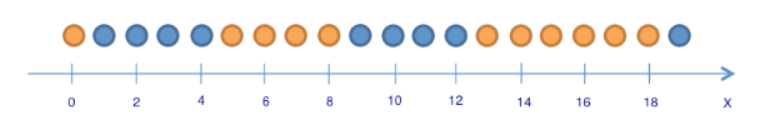
***1.4.1) Энтропия:***

*Информационная энтропия* — мера неопределённости некоторой системы, в нашем случае, непредсказуемость появления объекта конкретного класса в выборке. Чем выше энтропия, тем система более хаотичная.

В данной формуле - вероятность, того что выбранный объект принадлежит -му классу.

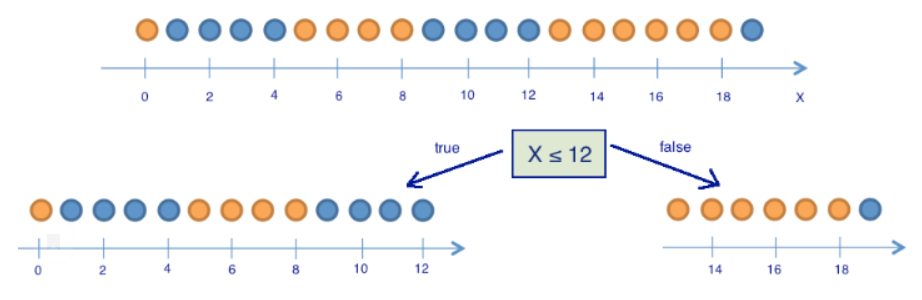
Пример:

Пусть мы хотим определять цвет шарика, основываясь на его координате и у нас имеется следующая обучающая выборка:



В выборке 9 синих и 11 желтых шариков. Если мы случайно вытащили шарик, то он с вероятностью будет синим и с вероятностью – желтым. Следовательно, энтропия состояния

Теперь проследим, как изменится энтропия, если разбить шарики на две группы – с координатой меньше либо равной 12 и больше 12. Получим следующее разбиение:



В левой группе оказалось 13 шариков, из которых 8 синих и 5 желтых. Энтропия состояния этой группы равна В правой же группе оказалось 7 шариков, из которых 1 синий и 6 желтых. Энтропия состояния правой группы равна

.

Теперь посчитаем прирост информации:

, где q – число групп после разбиения, – число элементов выборки, у которых признак Q имеет -ое значение. В нашем случае после разделения получилось две группы (q = 2) – одна из 13 элементов ( = 13), вторая – из 7 (N2 = 7). Прирост информации получился

Некоторые широко используемые алгоритмы построения решающих деревьев основываются на жадной максимизации прироста информации на каждом шаге. На каждом этапе выбирается признак, который разбивает выборку таким образом, чтобы прирост информации получился наибольшим из всех возможных вариантов. Далее процедура повторяется рекурсивно до тех пор, пока энтропия не окажется равной нулю или какой-то малой величине (в случае, если дерево идеально не подгоняется под обучающую выборку чтобы избежать переобучения).

***1.4.1) Индекс Джини (Gini Impurity):***

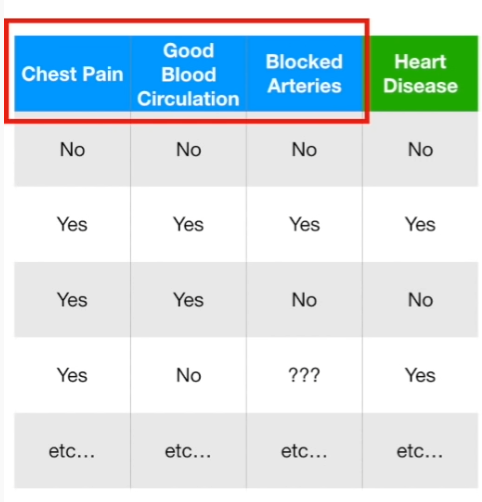
*Индекс Джини* – это величина, которая показывает, насколько хорошо в листе решающего дерева разделяются объекты по классам. С помощью индекса Джини определяется какое разбиение будет наилучшим в каждом узле дерева и надо ли вообще делать разбиения дальше.

Индекс Джини в узле вычисляется по формуле:

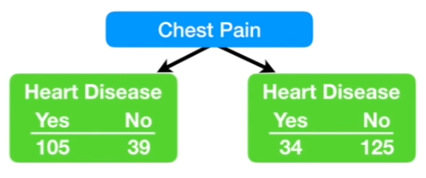
, где – вероятность появления объекта го класса в узле.

Пример:

Пусть у нас есть таблица с данными о сердечных заболеваниях 303 пациентов. У некоторых людей данные о каких-то признаках пропущены. Нам нужно понять какой признак из 3 имеющихся (Chest Pain, Good Blood Circulation, Blocked Arteries) наилучшим образом разделяет наши данные на каждом шаге построения дерева.



Посмотрим, как каждый признак разделяет наши данные на шаге разбиения корня и посчитаем индекс Джини для каждого разбиения:

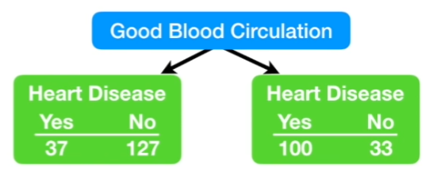


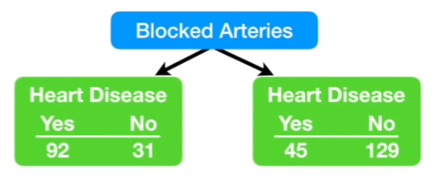
Для левого листа индекс Джини будет следующим:

Для правого листа:

Теперь, когда мы подсчитали индекс Джини для каждого листа, мы можем подсчитать индекс Джини для признака Chest Pain на данном шаге разбиения дерева. В левом листе 144 объекта, а в правом – 159, а не поровну, поэтому он будет равняться взвешенному среднему индексов Джини правого и левого листа:

Аналогично для оставшихся двух признаков:





Признак имеет наименьший индекс Джини, а значит он наилучшим образом разделяет пациентов.

Таким образом на каждом шаге разбиения дерева считается индекс Джини в уже имеющемся листе и индекс Джини для каждого признака, который может разделить объекты выборки в этом листе, далее выбирается признак с наименьшим индексом Джини или же если в самом листе индекс Джини наименьший, его больше не разбивают, так как дальнейшие разбиения делают наши данные более хаотичными и не приводят ни к каким улучшениям.

***1.5) Переобучение***

Переобучение происходит, если используется очень гибкая модель, которая просто запоминает обучающий набор данных, подгоняя условия в узлах непосредственно под него. Проблема в том, что такая модель помимо закономерностей в данных выявляет еще и любой присутствующий в них шум (какие-либо отклонения от закономерностей, исключения из правил). Такую гибкую модель обычно называют высоковариативной, так как параметры, формирующиеся в процессе обучения (такие как например условия в узлах дерева решений) будут сильно меняться в зависимости от обучающей выборки.

В то же время, недостаточно гибкая модель будет иметь большую погрешность, так как она делает слишком много предположений относительно тренировочных данных.

В обоих случаях — и при высокой погрешности, и при высокой вариативности — модель не сможет достаточно эффективно обрабатывать новые данные.

Дерево решений переобучается, если не ограничить его максимальную глубину. Такое дерево будет обладать неограниченной гибкостью и может разрастаться, пока идеально не подгонится под обучающую выборку. Каждому образцу из этой выборки будет соответствовать один лист. Если же ограничить глубину дерева, оно больше не будет идеально подстраиваться под обучающий набор данных, то есть в каждом листе будет больше чем один объект. Таким образом, за счет увеличения погрешности, уменьшается вариативность.

Также вместо ограничения глубины, которое приводит к уменьшению вариативности за счет увеличения погрешности, можно собрать множество деревьев в одну модель - ансамбль. Это и будет модель, с помощью которой я решал задачу предсказания успеваемости студентов – Случайный лес (Random Forest).

**2) Бэггинг**

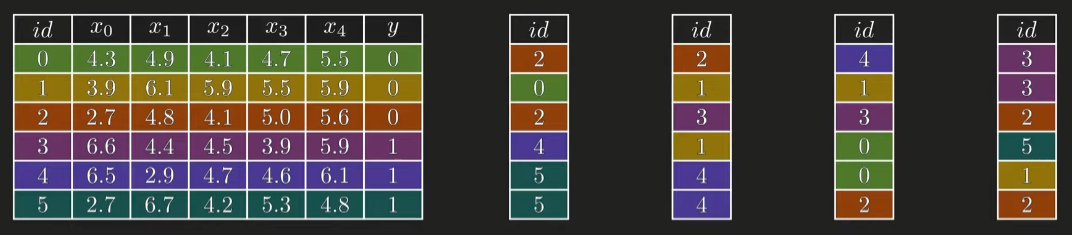
Случайный лес является примером ансамблевого алгоритма машинного обучения, а конкретно – примером бэггинга, объединяющего в себе решающие деревья, поэтому для начала стоит разобраться с тем, что такое бэггинг.

*Бэггинг* - от английского bagging (bootstrap aggregation). Идея бэггинга в том, что мы хотим подобрать и обучить несколько независимых моделей и усреднить их прогнозы для уменьшения вариативности (которая как раз приводит к переобучению). В бэггинге базовые алгоритмы принадлежат к одному и тому же семейству, они обучаются параллельно и почти независимо друг от друга, а финальные результаты получаются путем аггрегирования.

Чтобы понять, как исходный набор данных разбивается для формирования выборок для базовых алгоритмов, используется понятие бутстрэпа.

Процесс *бутстрэпа* представляет собой выбор n объектов с возвращением из обучающей выборки. Полученные с помощью бутстрэпа выборки будут использоваться для обучения слабых учеников. Такие выборки являются в значительной степени независимыми.

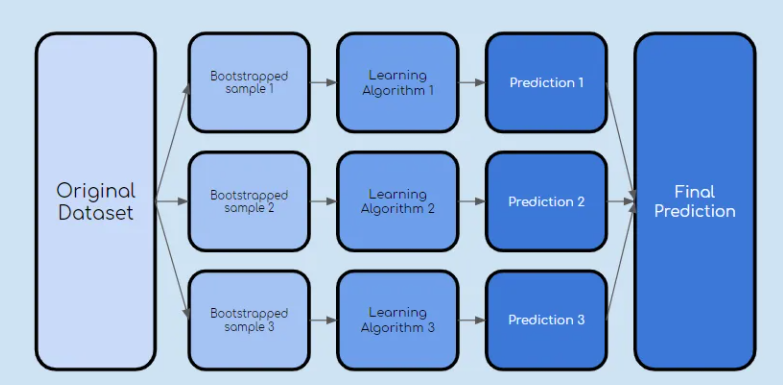
Вот пример 4 бутстрэп выборок из выборки, имеющей 6 объектов:



Объекты в бутстрэп выборках могут повторяться, так как объекты из обучающей выборки выбираются с возвращением. Следовательно, каждая выборка имеет только часть исходных данных задачи, что способствует независимому обучению базовых алгоритмов на бутстрэп выборках.

Алгоритм построения бэггинга следующий:

1. Исходный набор данных разделяется на n бутстрэп выборок.
2. На этих n выборках обучаются n однородных моделей.
3. Результаты, полученные моделями, аггрегируются чтобы получить конечный результат. В задачах классификации обычно просто выбирается класс, который встречается чаще остальных, а в задаче регрессии ответы моделей усредняются.



**3) Случайный Лес (Random Forest)**

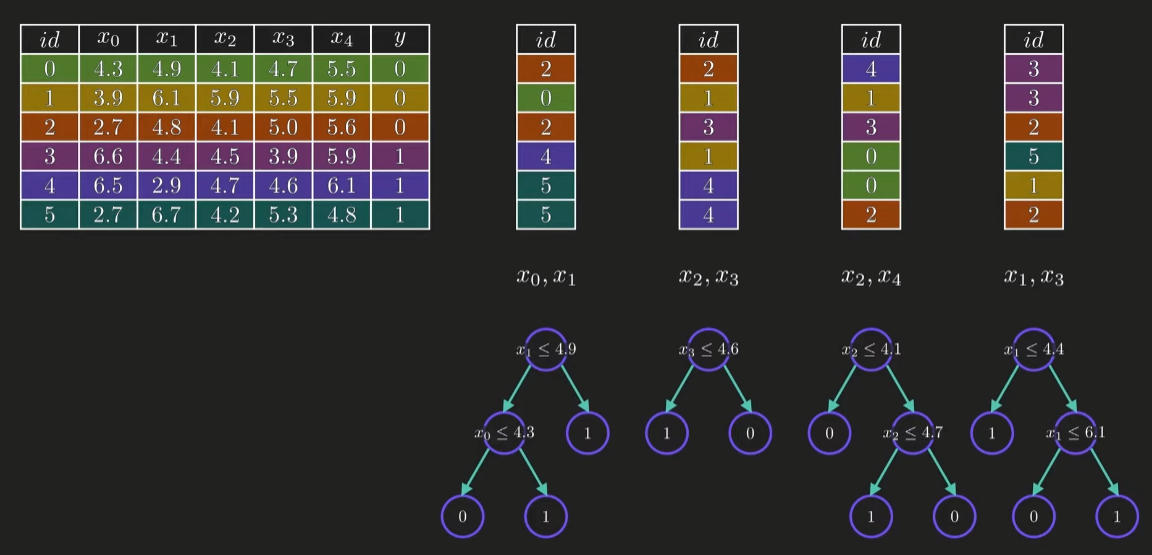
Самым распространенным примером бэггинга является случайный лес - модель, объединяющая в себе много решающих деревьев.

Первые работы по постоению ансамблей решающих деревьев были опубликованы в 1990-х годах. Чуть позднее был развит более строгий подход, ныне известный как метод случайных подпространств. Суть этого метода заключается в том, что для построения каждого отдельного дерева из всех признаков случайным образом выбирается лишь какое-то их фиксированное количество.

Наконец, американским математиком-статистиком Лео Брейманом была предложена модель, названная случайным лесом. Такой случайный лес – это ансамбль решающих деревьев, которые строятся на основе бутстреп выборок из начальной обучающей выборки, при чем каждому дереву соответствует своя собственная бутстрэп выборка. Также в этой модели при разбиении каждой вершины используется лишь часть случайно выбираемых признаков. Классификация происходит с помощью голосования: за ответ принимается наиболее часто предсказываемый класс. В качестве ответа для задачи регрессии выбирается среднее между ответами всех деревьев.

Алгоритм построения и использования случайного леса следующий:

* + 1. Для *i = 1, 2, . . . ,* *K* (где K — количество деревьев решений в ансамбле) выполнить:
* Сформировать бутстрэп выборку D размера m по исходной обучающей выборке ;
* По бутстрэп выборке D построить решающее дерево, рекурсивно выполняя следующие шаги:
  + - * 1. из начальных n признаков случайно выбрать q признаков
        2. из q признаков найти признак, который наиболее эффективно разбивает выборку
        3. разбить выборку в текущей вершине на две части
    1. После выполнения шага 1 имеем ансамбль решающих деревьев
    2. Предсказание ответа для новых наблюдений делается следующим образом:
       - 1. для регрессии:
         2. для классификации выбирается наиболее часто встречающийся среди предсказанных деревьями класс.



На изображении 4 бутстрэп выборки, каждой из которых соответствует решающее дерево, обученное на ней. Видно, что деревья получились разными.

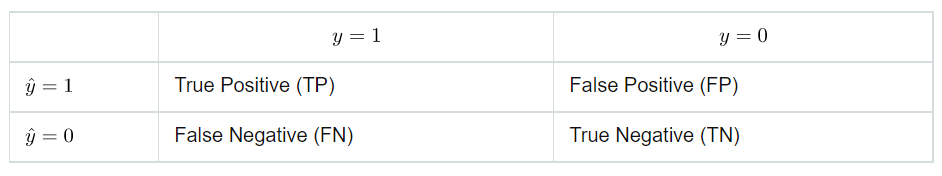
Случайные леса обладают целым рядом преимуществ, а именно:

1. Точность случайных лесов сильно повышается по сравнению с отдельными решающими деревьями, так как засчет бэггинга и использования метода случайных подпространств при разбиении вершин деревья в ансамбле в большой степени независимы.
2. Проблема переобучения отсутствует, даже если признаков больше чем объектов в обучающем наборе данных. Так как при разбиении каждой вершины, выбираются лишь некоторые признаки, не приходится решать сложную задачу изначального отбора наиболее важных признаков, что необходимо для многих других ансамблевых моделей.
3. Случайный лес довольно просто применять, так как единственными параметрами модели являются количество деревьев в ней и количество признаков, случайно выбираемых для разбиения в каждой вершине дерева.

## **4) Метрики для сравнения алгоритмов**

В ходе решения задачи предсказывания успеваемости студентов, мне нужно было как-то оценивать эффективность моделей. Для этого важно понимать, как работают некоторые метрики в машинном обучении, а именно *Accuracy*, *Precision* и *Recall*.

Перед тем, как перейти непосредственно к метрикам, необходимо ввести важный объект для описания этих метрик в терминах ошибок классификации — матрицу ошибок (confusion matrix). Допустим, что у нас есть два класса и алгоритм, который предсказывает принадлежность каждого объекта одному из классов. Тогда матрица ошибок классификации будет выглядеть так:



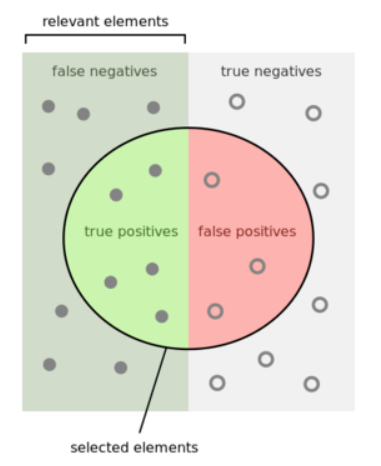
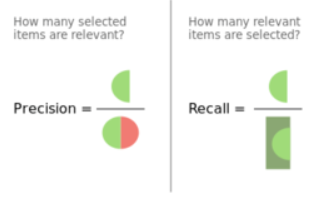
Здесь — это ответ алгоритма на объекте, а — истинный класс объекта. Таким образом, есть два вида ошибок классификации: False Negative (FN) и False Positive (FP).

Итак, перейдем к самим метрикам:

1) Accuracy — это доля правильных ответов алгоритма.

2) Precision (точность) можно понимать, как долю объектов конкретного класса среди всех элементов, которые алгоритм отнес к этому классу. Например, Precision для класса 1 будет вычисляться так:

3) Recall (полнота) показывает, какую долю объектов конкретного класса из всех объектов этого класса идентифицировал алгоритм. Например, Recall для класса 1 будет вычисляться так:

## **5) Предсказывание успеваемости студентов**

Для решения основной задачи, то есть предсказывания успеваемости студентов на основе данных опроса, я использовал язык программирования Python и его библиотеку Sci-Kit Learn. Также мне потребовалась библиотека Pandas, в частности DataFrame из этой библиотеки, чтобы удобно хранить данные в виде таблицы и подготавливать их для обучения моделей. Также я использовал библиотеку Openpyxl, чтобы получать информацию из Excel-таблиц успеваемости студентов с сайта ВШЭ.

***5.1) Препроцессинг:***

Из всех студентов в таблице я отобрал студентов факультета ИМиКН, и оставил только тех, чью успеваемость я мог проверить с помощью рейтингов с сайта ВШЭ. Всего получилось 240 студентов.

Первым делом я создал pandas.DataFrame из вопросов опроса студентов, извлёк данные о наличии неудовлетворительных оценок из таблиц с сайта ВШЭ и добавил колонку “Наличие неудовлетворительных оценок” в мою таблицу. Таким образом я получил классический датасет для машинного обучения: матрицу признаков (X) и вектор-столбец ответов (y).

Далее, чтобы модель смогла обучиться на таком датасете, я сделал все категориальные признаки числовыми: ответы “да” и “нет” поменял на 0 и 1, к признакам, у которых больше двух различных значений я применял функцию *get\_dummies()*. Таким образом я преобразовал исходный датасет с категориальными признаками в полностью готовый для обучения модели датасет, состоящий только из цифр.

***5.2) Обучение моделей:***

Чтобы проверить, насколько хорошо алгоритм случайного леса подходит для данной задачи, я решил помимо случайного леса также обучить решающее дерево и модель на основе алгоритма градиентного бустинга. Решающее дерево я выбрал, чтобы понять, имеет ли смысл использовать такое усложнение над деревьями как случайный лес или же одного дерева достаточно для решения этой задачи, а градиентный бустинг я выбрал так как этот алгоритм – один из самых широко используемых алгоритмов в машинном обучении.

***5.3) Сравнение моделей:***

Для сравнения трех вышеописанных алгоритмов я использовал 3 метрики: Accuracy, Precision и Recall.

Случайный лес каждый раз генерирует разные деревья решений, поэтому и результаты метрик при каждом обучении получаются разными, естественно я обучал лес до тех пор, пока не получил лучший результат. Для бустинга и решающего дерева я методом проб перебирал параметры, такие как глубина дерева для обоих алгоритмов и количество деревьев для бустинга, чтобы получить наиболее точные предсказания.

Наилучшие результаты получились следующими:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Accuracy | Precision (0) | Precision (1) | Recall(0) | Recall(1) |
| Random Forest | 0.8125 | 0.9412 | 0.7419 | 0.6667 | 0.9583 |
| Decision Tree | 0.8125 | 0.8260 | 0.8000 | 0.7917 | 0.8333 |
| Gradient Boosting | 0.8125 | 0.9412 | 0.7419 | 0.6667 | 0.9583 |

Из таблицы видно, что процент правильных ответов (Accuracy) одинаковый у всех трёх моделей.

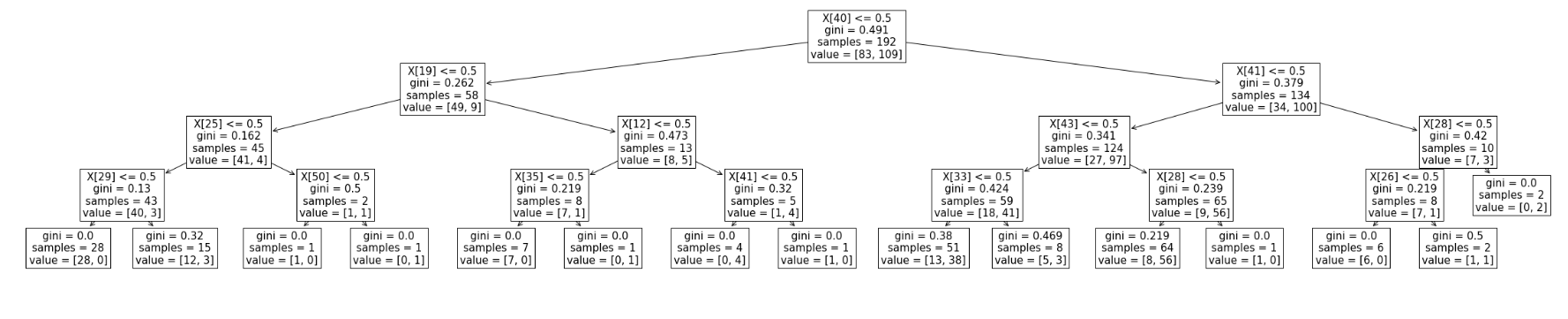
Однако задача состоит в том, чтобы по данным опроса студента, предсказать будет ли у него неудовлетворительная оценка, и поэтому наиболее важно пропустить как можно меньше студентов с потенциальными неудовлетворительными оценками. Иными словами, по моему мнению, главной целью задачи является как можно лучше научиться находить именно студентов с неудачными сдачами экзаменов в перспективе, при том, что определять студентов, у которых в учебе все будет успешно – менее важно.

Именно поэтому полнота для класса 1 (recall(1)) является здесь наиболее важной метрикой, по моему мнению, ведь она показывает, какую долю объектов первого класса из всех объектов этого класса нашел алгоритм. Значение этой метрики наиболее высокая у модели на основе алгоритма случайного леса и равна значению у модели на основе градиентного бустинга, как и значения всех остальных метрик. Скорее всего, если бы выборка была в несколько раз больше, модели не показывали бы одинаковые результаты на всех метриках, но тем не менее можно сделать вывод, что обе эти модели довольно хорошо справляются с задачей.

# **Вывод**

Итак, в своей работе я сравнил 3 алгоритма на выборке из 240 студентов. По значениям метрик можно сказать, что для решения конкретно этой задачи лучше подходят сложные ансамблевые алгоритмы, в моем случае случайный лес и градиентный бустинг. Эти алгоритмы по метрике recall(1) справились лучше, чем такая базовая модель, как решающее дерево, хотя процент правильных ответов у всех трёх алгоритмов был одинаков.

Также можно выделить наиболее важные признаки, по которым наилучшим образом разделяется выборка. Сделать это можно с помощью визуализации решающего дерева, у которого глубина ограничена. В моем случае я ограничил глубину до 4:



На первых четырех уровнях дерево разделяет выборку по следующим признакам:

1. X[12]: "Образование Ваших родителей?\_Среднее (школа)"
2. X[19]: "На какие средства Вы живете?\_Деньги дают родные"
3. X[25]: "Каковы условия Вашего проживания?\_Живете с родителями/родными/супругом"
4. X[26]: "Сколько времени Вы уделяете самостоятельной подготовке к занятиям (в среднем)?\_7 часов в неделю"
5. X[28]: "Сколько времени Вы уделяете самостоятельной подготовке к занятиям (в среднем)?\_Все свободное время"
6. X[29]: "Сколько времени Вы уделяете самостоятельной подготовке к занятиям (в среднем)?\_Готовлюсь только перед занятиями"
7. X[33]: "Как много Вы пропускаете аудиторных занятий?\_Среднее количество пропусков"
8. X[35]: "Какая у Вас семья?\_Оба родителя вместе"
9. X[40]: "Бывают ли у Вас долги по экзаменам/зачетам?\_Всегда"

10) X[41]: "Бывают ли у Вас долги по экзаменам/зачетам?\_Нет"

11) X[43]: "С какими оценками Вы закончили школу?\_4-5"

12) X[50]: "Участвуете ли Вы в олимпиадах?\_Нет" Значений каждого из этих признаков всего два: 0 (нет) и 1 (да).

Другими словами, наилучшим индикатором для прогнозирования успеваемости являются ответы студентов на следующие вопросы:

1. Образование Ваших родителей?
2. На какие средства Вы живете?
3. Каковы условия Вашего проживания?
4. Сколько времени Вы уделяете самостоятельной подготовке к занятиям (в среднем)?
5. Как много Вы пропускаете аудиторных занятий?
6. Какая у Вас семья?
7. Бывают ли у Вас долги по экзаменам/зачетам?
8. С какими оценками Вы закончили школу?
9. Участвуете ли Вы в олимпиадах?

Результаты вполне ожидаемы и понятны, так как ответы на эти вопросы дают понять, насколько хорошо учился студент раньше, насколько ответственно он относится к ней на момент опроса (пропускает ли занятия и как много времени уделяет самоподготовке), есть ли у него возможность посвящать большую часть времени учебе или же он работает, насколько благополучная у него семья. Все эти факторы безусловно влияют на успеваемость студента в той или иной мере.

Хоть случайный лес и градиентный бустинг и не предсказывают со стопроцентной точностью правильную метку класса студента, они распознают большую часть студентов, получивших неудовлетворительные оценки, что является главной целью данной задачи. Таким образом, используя алгоритмы машинного обучения, можно распознавать студентов, склонным к неудачам в учебе и предлагать им различные факультативы и дополнительные занятия, чтобы повысить общую успеваемость.

# **Список литературы**

1. <https://cyberleninka.ru/article/n/sluchaynye-lesa-obzor>
2. <https://ml-handbook.ru/chapters/decision_tree/intro>
3. <https://habr.com/ru/company/ods/blog/322534/>
4. <https://statquest.org/>
5. <https://www.youtube.com/channel/UC7Fs-Fdpe0I8GYg3lboEuXw>
6. <https://cyberleninka.ru/article/n/zadachi-i-metody-intellektualnogo-analiza-obrazova>
7. <https://ml-handbook.ru/chapters/model_evaluation/intro>
8. <https://habr.com/ru/company/ods/blog/324402/>
9. <https://tproger.ru/translations/python-random-forest-implementation/>

10. <https://neurohive.io/ru/osnovy-data-science/ansamblevye-metody-begging-busting-i-steking/>

11. <https://scikit-learn.org/stable/index.html>