Федеральное государственное образовательное бюджетное учреждение высшего образования

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

Факультет информационных технологий и анализа больших данных

Департамент анализа данных и машинного обучения

Выпускная квалификационная работа

на тему «Сравнительный анализ ансамблевых методов деревьев решений: бэггинг, бустинг, случайные леса»

Направление подготовки 01.03.02 «Прикладная математика и информатика»

Выполнил студент учебной группы ПМ18-1

Изображение выглядит как текст

Автоматически созданное описаниеАблязова Алия Маратовна

Руководитель д.э.н., профессор

Демин Игорь Святославович

**ВКР соответствует предъявляемым требованиям**

Руководитель Департамента

д.э.н., профессор

В.И. Соловьев

«26» мая 2022 г.

Москва - 2022 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ……………………………………………………………………..…3

Глава 1. Инструментарий деревьев решений……………………………………5

* 1. Элементы дерева решений……………………………………………………5
  2. Построение дерева решений классификации и регрессии…………………7
  3. Ансамблевые методы. Бэггинг и случайный лес…………………….…….11
  4. Ансамблевые методы. Бустинг………………………………………….….14

Глава 2. Прикладные задачи для анализа и метрики оценки результатов.......23

* 1. Прикладная задача регрессии………………………………………………25
  2. Прикладная задача классификации………………………………………...30
  3. Метрики оценки результатов……………………………………………….38

Глава 3. Построение моделей и сравнение ансамблевых методов...................42

* 1. Исследуемые ансамблевые модели и их параметры....................................42
  2. Построение моделей и получение результатов в задаче регрессии...........46
  3. Построение моделей и получение результатов в задаче классификации..49
  4. Сравнение результатов и метрик оценки результатов.................................54

ЗАКЛЮЧЕНИЕ.....................................................................................................55

СПИСОК ИСТОЧНИКОВ....................................................................................58  
ПРИЛОЖЕНИЕ ....................................................................................................60

ВВЕДЕНИЕ

Деревья решений являются распространенным методом машинного обучения при решении задач, однако при своей хорошей интерпретируемости могут быть не самыми эффективными. Ансамблевые методы деревьев решений есть инструмент анализа данных и машинного обучения, которые комбинируют несколько базовых алгоритмов, чтобы построить одну оптимальную модель предсказаний. Такие методы могут быть сложнее и дольше для построения, однако использование ансамблей деревьев решений помогает достичь лучших результатов. Ансамблевые методы, как правило, производят более точные предсказания и могут решать более широкий круг задач, чем однокомпонентная модель. Данные методы также применяются, чтобы уменьшить дисперсию предсказанных значений.

Ансамбли используются как для решения задач классификации, так и для решения задач регрессии. Существует множество областей применения ансамблевых методов в разных сферах, в том числе в сфере информационной безопасности. В данной работе исследуется эффективность различных методов в задачах, относящихся к данной области, и для полноты сравнительного анализа рассматриваются две прикладные задачи регрессии и классификации. Таким образом, актуальность данной работы состоит в исследовании ряда ансамблевых методов на задачах, имеющих специфику области информационной безопасности. Исследование поможет определить, какие ансамблевые модели можно применять для получения наилучших результатов в рамках задач рассматриваемой области.

Выпускная квалификационная работа посвящена сравнительному анализу ансамблевых методов деревьев решений на двух задачах из области кибербезопасности. Работа состоит из введения, трех глав, заключения, списка литературных источников и приложения. В первой главе предоставлен теоретический материал об инструментах деревьев решений и ансамблевых методов, от алгоритма построения однокомпонентной модели дерева решений до самих ансамблевых методов и их особенностей. Вторая глава посвящена описанию прикладных задач и особенностей используемых наборов данных, а также описанию используемых метрик оценки результатов. В третьей главе реализуется построение ансамблевых моделей рассматриваемых методов и проводится сравнительный анализ полученных метрик эффективности.

Таким образом, объектом исследования заданной темы являются ансамблевые методы деревьев решений, а предметом - особенности каждого метода и результаты работы построенных моделей на указанных прикладных задачах.

Целью работы является сравнение ансамблевых методов деревьев решений при решении задач регрессии и классификации и исследование метрик эффективности работы построенных моделей.

Задачами данной работы являются анализ инструментария ансамблевых методов деревьев решений, исследование прикладных задач регрессии и классификации в области кибербезопасности, проведение первоначального анализа данных и подготовки данных для моделей, определение используемых метрик эффективности работы методов, построение моделей и сравнительный анализ результатов их работы.

Исследование ансамблевых методов деревьев решений и их сравнительный анализ поможет определить наилучшую модель в зависимости от специфики поставленной прикладной задачи и используемых данных.

Глава 1. Инструментарий деревьев решений

* 1. Элементы дерева решений

Деревья решений – инструмент анализа данных, позволяющий решать задачи классификации и регрессии. Деревья решений часто используются в анализе данных и машинном обучении, потому что позволяют разбить сложные данные на более простые части, а также являются одними из наиболее легкими алгоритмами для интерпретации и понимания. Алгоритм дерева решений относится к типу алгоритмов с учителем. Целью использования дерево решений является создание модели, которая бы предсказывала класс или значение целевой переменной, создав множество простых методов принятия решений. Само дерево решений представляет собой метод, структурно состоящий из двух основных типов элементов: узлы и листья. Кроме этих элементов, существуют и другие понятия, относящиеся к терминологии деревьев решений:

1. Корневой узел, который представляет всю обучающую или тестовую выборку и при дальнейшем построении алгоритма делится на два и более подмножеств элементов исходной выборки
2. Разбиение/разделение/сплит – процесс разделения узла в два и более подузлов
3. Узлом решения называется узел, который далее можно разделить на другие узлы
4. Листьями (конечным узлом) называются узлы, которые невозможно разделить на последующие узлы, например, если в них присутствует один элемент выборки
5. Отсечением ветвей называется избавление от узлов решения. Отсечение, по своей сути, есть явление, противоположное разбиению/сплиту
6. Родительским узлом называется узел, который разбивается в подузлы. В свою очередь, подузлы называются дочерними

Родительский узел

Корневой узел

Узел решения

Дочерний узел

Узел решения

Лист/конечный узел

Лист/конечный узел

Лист/конечный узел

*Схема1. Элементы дерева решений*

Каждый узел представляет собой тестирование атрибута обучающего множества на удовлетворение условию, каждая ветвь - результатом тестирования, а каждый лист отвечает за присваиваемый атрибуту класс в случае задачи классификации или значение целевой переменной для задачи регрессии. При проверке соответствия условию в первом (корневом) узле первичное множество делится на два или более подмножеств, которые проверяются в дальнейших узлах и далее, пока атрибуты последних подмножеств не достигнут листьев дерева с присеваемым классом или интервалом.

Обучение алгоритма деревья решений есть непосредственное построение дерева, что заключается в рекурсивном делении обучающего множества признаков на подмножества с некими условиями в узлах.

Листья дерева появляются в случае, если в подмножестве остаются один или более признаков одного класса или заданы иные параметры остановки построения дерева, например, максимальная глубина.

* 1. Построение дерева решений классификации и регрессии

Построение дерева решений различается в зависимости от решаемой задачи и, соответственно, дискретной или непрерывной целевой переменной. В данной работе исследуется как задача классификация, так и задача регрессии, поэтому рассмотрим алгоритмы построения одного дерева решений для решения каждого типа.

Классификация:

При решении задачи классификации в каждом узле находится условие: если условие рассматривает категориальный признак, то выборка делится на подмножество, данный признак которого соответствует категории в узле, и подмножество переменных, чья категория не соответствует указанной в узле; если рассматриваемый атрибут является не категориальным, а непрерывным, то условие в узле проверяет принадлежность признака переменной к указанному промежутку.

После разделения в узле исходной выборки на подвыборки, необходимо посчитать метрику “чистоты” разделения. Наиболее распространенными метриками “чистоты” разбиения являются энтропия, которую можно интерпретировать как количество информации, необходимой для верного описания элемента данных, и индекс Джини, который можно интерпретировать как функцию потерь, которая используется для исчисления качества разделений датасета при построении дерева и показывает вероятность случайно выбранного атрибута провести неверную классификацию. Для бинарной классификации, которая исследуется в данной работе, индекс Джини является наиболее распространенной метрикой «чистоты» разбиения, поэтому он будет рассматриваться подробнее.

Значение индекса Джини принадлежит промежутку [0;1], где 0 сообщает о полностью верной классификации, а значение индекса 0.5 говорит о равном распределении элементов по классам. Данный индекс вычисляется путем вычитания из единицы суммы возведенных в квадрат вероятностей каждого класса:

,

Где N – количество классов и p(i) – вероятность классификации элемента набора данных в класс i.

Если в узле рассматривается не категориальный, а непрерывный признак, то промежуток в условии выбирается по следующему алгоритму: значения атрибута сортируются по возрастанию, и строится n-1 (n - количество записей набора данных) деревьев, где числом, относительно которого делится выборка, является среднее арифметическое между значениями признака каждой последующей пары переменных набора данных. Для каждого дерева находится индекс Джини, по которому определяется подходящее число для определения промежутка условия.

Рассчитав индекс Джини для каждого признака набора данных, можно начать построение дерева, в корневом узле расположив признак, чье разделение данных имеет наименьший индекс Джини. Дальнейшее построение дерева происходит по такому же алгоритму, где каждый последующий узел решения попадает в модель путем выбора разбиения с наименьшим индексом Джини.

Построение дерева заканчивается в тот момент, когда образуются листья (конечные узлы), то есть достигается неделимое число элементов набора данных, либо количество элементов меньше указанного наименьшего значения подвыборки. Лучше практикой считается задание параметра наименьшего значения подвыборки при разбиении, так как остановка построения дерева в случае неделимости выборки может привести к переобучению модели.

Регрессия:

В задаче регрессии целевая переменная является непрерывной, значит, в конечных листьях дерева, содержащих предсказание для каждой записи набора данных, находится число. При выборе числа, относительно которого происходит разделение выборки в узле, происходит последовательное вычисление и сравнение суммы квадратичных отклонений.

На первом этапе в условии узла находится среднее арифметическое значение между наименьшим значением целевой переменной при рассматриваемом признаке и последующим значением целевой переменной этого признака. Таким образом, значения целевой переменной делятся на две подвыборки/кластера.

Предсказанное значение для каждого кластера есть среднее арифметическое между всеми значениями в данной подвыборке. Далее для каждого целевого значения вычисляется квадратичное отклонение от предсказанного значения по формуле:

,

где - предсказанное разбиением значение целевой переменной,

– истинное значение целевой переменной из обучающей выборки.

Метрикой эффективности для разбиения такого дерева является сумма квадратичных отклонений:

.

По этому алгоритму вычисляются суммы квадратичных отклонений для каждого среднего арифметического по всем значениям рассматриваемого атрибута.

Узел, чье разбиение имеет наименьшую сумму квадратичных отклонений, является оптимальным корневым узлом регрессионного дерева решений.

Если в узле находится значение не непрерывного признака, а категориального, то отличие алгоритма будет в том, что кластеризация целевых значений будет происходить по категориям рассматриваемого признака.

Таким образом, по аналогии с выбором корневого узла, также выбираются и последующие узлы, чье разбиение дает наименьшую сумму квадратичных отклонений.

Построение регрессионного дерева решений прекращается, то есть образуются листья дерева в тот момент, когда на сплите выборка неделима или же в подвыборках находится элементов меньшего количества, чем указано при задании параметров дерева. Как уже было упомянуто выше, выставление параметра наименьшего количества элементов подвыборки помогает бороться с переобучением.

Деревья решений, хоть и являются довольно простыми в построении, а также наиболее интерпретируемыми модели, часто уступают в эффективности и точности предсказаний другим моделям, в частности, потому что обладают проблемой высокой дисперсией предсказанных значений. Ансамблевые методы деревьев решений являются решением данной проблемы, так как агрегирование результатов множества деревьев решений путем их усреднения снижает дисперсию и способствует более точным предсказаниям модели.

* 1. Ансамблевые методы. Бэггинг и случайный лес

Ансамблевые модели — это собирательный термин для подходов машинного обучения, где несколько моделей обучаются вместе для решения одной задачи и далее объединяются, чтобы повысить эффективность решения. Как правило, для обучения используется “однородная” модель ансамбля, основывающийся на определенном базовом алгоритме обучения, также имеющая название «слабый ученик». В данной работе базовым алгоритмом обучения является дерево решений для каждого ансамблевого метода. В этом случае ансамблевые модели позволяют, взяв за основу классификационное или регрессионное дерево решений, определить, какие использовать признаки из обучающей выборки или условия в узлах решений деревьев, и найти значение целевой переменной (класс или число), основываясь на агрегированных результатах базовых деревьев решений.

Самыми распространенными видами ансамблевых методов деревьев решений являются стэкинг, бэггинг, бустинг и случайные леса, из которых в работе будут рассмотрены бэггинг, случайные леса и бустинг.

Бэггинг (Bootstrap Aggregation) - алгоритм, который собирает предсказания всех моделей ансамбля для получения более точных результатов. Бэггинг позволяет бороться с большой дисперсией результатов.

При бэггинге каждое дерево обучается независимо друг от друга, что и позволяет получить наиболее близкий к правде результат. Независимость деревьев в модели бэггинга обеспечивает метод генерации выборок для каждого базового ученик, называющийся бутстрэпинг.

Бутстрэпинг подразумевает под собой метод создания выборок из исходного датасета с заменой (resampling), то есть каждый случайным образом выбранный элемент, попадающий в обучающую выборку для отдельного дерева решений, остается в исходной выборке и может быть случайно выбран снова.

Таким образом бутстрэпинг обеспечивает и более полное представление о популяции датасета, так как вероятность попадания в выборку элемента, который встречается в исходной выборке чаще других, больше. Кроме того, метод бутстрэпинга помогает бороться с несбалансированностью набора данных в случае, если при классификации распределение количества элементов, принадлежащих к целевым классам, сильно неравное. N выборок, созданные методом бутстрэпинга попадают в n базовых алгоритмов (деревьев) и обучаются независимо, позволяя получить результаты, наилучше отображающие общую закономерность набора данных, и бороться с переобучением алгоритма.

После генерации выборок методом бутстрэпинга, строятся n базовых алгоритмов (деревьев решений), каждое из которых дает определенный результат для каждого входящего элемента обучающей или тестовой выборки.

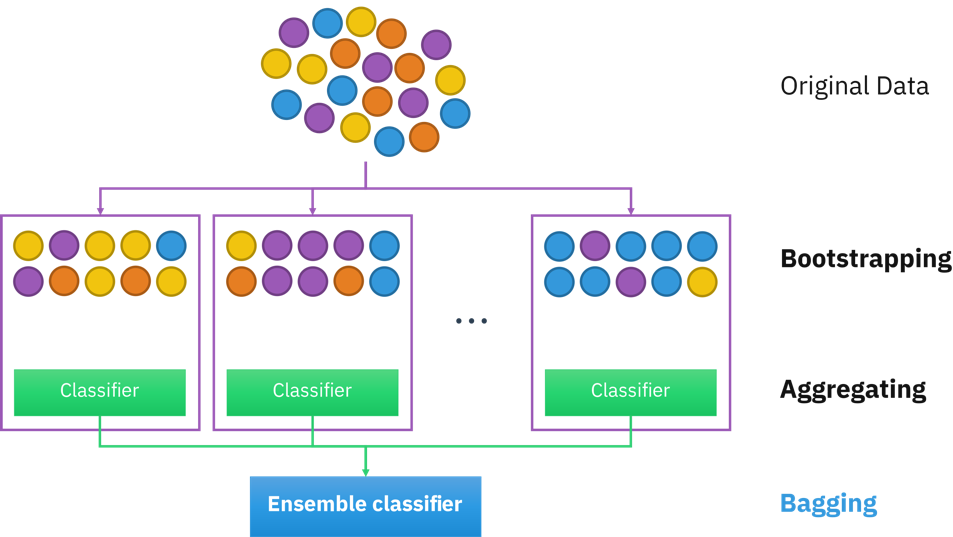


Рисунок 1. Бутстрэппинг и бэггинг

Результатом работы бэггинга являются агрегированные результаты по всем базовым алгоритмам, то есть в случае решения задачи регрессии это среднее арифметическое значение по всем деревьям, а при классификации класс, набравший наибольшее количество голосов, то есть наиболее чаще встречающийся среди предсказаний деревьев решений.

Одной из разновидностей метода бэггинга является довольно эффективный и часто используемый алгоритм случайного леса. Случайный лес обеспечивает еще большую независимость базовых алгоритмов за счет модифицированного алгоритма генерации обучающих выборок.

Обучающие выборки для каждого дерева решений в данном методе так же формируются методом бутстрэп из исходной популяции, однако модель случайного леса отличается от бэггинга тем, что набор признаков, которые может использовать каждый базовый алгоритм является разным и не равен множеству всех атрибутов обучающей выборки. То есть деревья решений не одинаковы, и на каждом разделении находится оптимальный узел решений, атрибут которого может быть выбран из случайного набора признаков, в то время как в бэггинге, каждое дерево может выбрать атрибуты из столбцов всей обучающей выборки. Такой подход помогает декоррелировать деревья, то есть обеспечить еще большую их независимость.

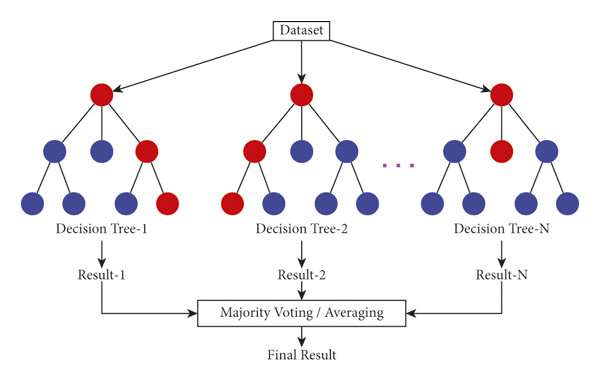


Рисунок 2. Случайный лес

Результатом работы случайного леса, как подвида метода бэггинга также являются среднее арифметическое значение по всем построенным деревьям или чаще встречающийся класс в задачах регрессии и классификации соответственно.

* 1. Ансамблевые методы. Бустинг

Помимо бэггинга в данной работе также рассматривается ансамблевый метод под названием бустинг. Данный алгоритм используется для снижения смещения результатов, то есть математического ожидания разности между предсказанным значением и истинным значением. Бустинг меняет алгоритм, чтобы с каждой новой моделью минимизировать ошибки предыдущей. В отличие от бэггинга, где базовые алгоритмы обучаются независимо, в этом методе модели обучаются последовательно, и каждая новая модель фокусируется на неверных предсказаниях модели предыдущей, например, применяя веса и увеличивая вес тех элементов набора данных, которые в большей степени повлияли на ошибочное предсказание. Наиболее часто в практике используют следующие виды бустинга: адаптивный бустинг и градиентный бустинг.

В адаптивном бустинге “слабым учеником” является дерево решений регрессии или классификации, которое состоит из одного узла с разбиением и двумя листьями, называемого “пень” (stump). Такое простейшее дерево может использовать только одну переменную для принятия решения. Адаптивный бустинг использует веса для каждого элемента набора данных, чтобы определить, насколько важным является верная классификация или предсказание непрерывного значения отдельного элемента для всего алгоритма.

На первом шаге все элементы набора данных имеют одинаковые веса, то есть все элементы набора данных считаются одинаково важными:

Где N – количество элементов датасета.

Сумма всех весов всегда равна единице, поэтому значение каждого отдельного веса лежит в промежутке от 0 до 1.

После получения результатов работы “слабого ученика” вычисляется индекс Джини для рассматриваемого разбиения. Если это первая итерация, то узел, чье разбиение имеет наименьший индекс, будет выбран корневым узлом модели.

Далее необходимо вычислить, какое влияние этот базовые алгоритм будет иметь при финальной классификации или регрессии. Это значение называется «Amount of Say» и вычисляется по формуле:

где TotalError для базового алгоритма есть сумма весов неверно классифицированных или попавших в неверный кластер значений элементов.

Amount of Say принимает положительное значение при лучших результатах и отрицательное значение при большем количестве неверно классифицированных элементов.

Далее алгоритм бустинга увеличивает веса элементов, которые были неверно классифицированы или кластеризованы и способствовали увеличению ошибки. А так как сумма всех весов должна быть равно 1, то веса верно классифицированных или кластеризованных элементов будут уменьшены. Формула ниже для обновления весов неверно предсказанных элементов набора данных:

,

где i – номер итерации,

– новый вес элемента,

– предыдущий вес элемента.

Аналогично, обновление весов верно предсказанных элементов происходит по формуле:

,

где i – номер итерации,

– новый вес элемента,

– предыдущий вес элемента.

Другими словами, вес элемента увеличится, если Amount of Say будет отрицательным, то есть элемент был неверно классифицирован или кластеризован, но уменьшится, если предсказание было верным.

При обновлении весов необходимо их нормализовать, поделив на новую вычисленную сумму весов. Таким образом, обновленные нормализованные веса становятся весами элементов при построении следующего «слабого ученика».

Последующие узлы также выбираются с помощью вычисления индекса Джини, а то, насколько важным будет классификация или предсказание значения на данной сплите, определяется с помощью переменной Amount of Say. На каждом шаге веса набора данных обновляются, что позволяет модели адаптивного бустинга оптимизировать результаты, наблюдая за теми элементами, которые способствуют ошибке предсказания.

Алгоритм прекращает работу, когда достигнуто максимальное заданное число итераций или все элементы верно классифицированы или предсказаны. Таким образом, адаптивный бустинг помогает увеличить точность базовых слабых алгоритмов, таких как дерево решений. Однако нужно иметь в виду, что адаптивный бустинг очень чувствителен к данным с шумами и выбросами, так как именно они в большей степени способствуют неверным предсказаниям целевой переменной, поэтому при предварительной обработке данных необходимо убрать такие элементы.

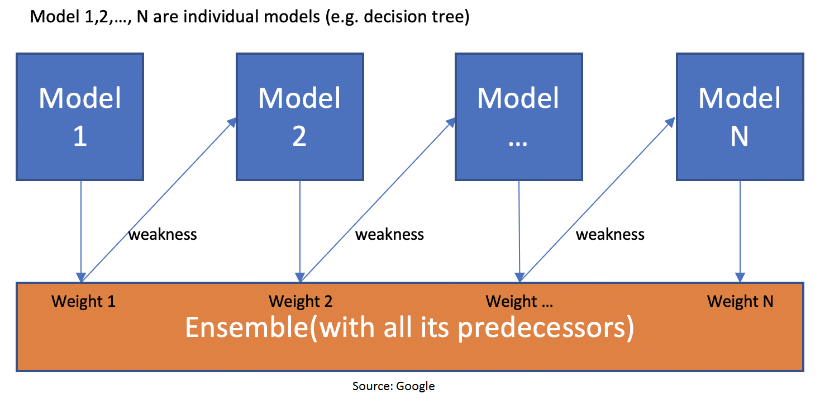


Рисунок 3. Адаптивный бустинг

Следующий рассматриваемый алгоритм бустинга основывается на методе градиентного спуска и называется градиентный бустинг. Для реализации данного алгоритма необходимо вычисление и оптимизация функции потерь - чаще всего в случае регрессии это среднеквадратичная ошибка:

,

Где N – количество элементов набора данных,

– истинные значения целевой переменной,

– предсказанные значения целевой переменной.

В случае классификации же часто для оптимизации используется логарифмическая функция потерь, которую также называют бинарной кросс-энтропией:

,

Где N – количество элементов набора данных,

– истинные значения целевой переменной,

– вероятность верного предсказания для i-го элемента.

“Слабыми учениками” в градиентном бустинге также выступают деревья решений. Деревья добавляются друг к другу последовательно, и существующие в модели деревья не меняются. С помощью метода градиентного спуска решается задача минимизации функции потерь с каждом новым добавленным деревом.

Окончание построения алгоритма происходит в тот момент, когда функция потерь достигает приемлемого значения, или добавление новых деревьев не делает модель лучше.

Градиентный бустинг реализуется несколькими методами в библиотеках машинного обучения. В данной работе рассматриваются метод XGBoost и LightGBM.

XGBoost (Extreme Gradient Boosting) является оптимизированной модификацией метода градиентного бустинга и реализует параллелизацию построения деревьев решений. Параллелизация процесса помогает уменьшить время обучения модели.

Вместо того, чтобы обучать наилучшую возможную модель на исходных данных (как в большинстве используемых в машинном обучении методах), алгоритм обучает множество моделей на разных подвыборках тренировочной выборки, и затем голосованием определяется модель с наилучшими результатами.

Во многих случаях XGBoost работает лучше обычных алгоритмов градиентного бустинга. Библиотека Python также позволяет настраивать большое количество параметров модели для получения более точных результатов.

Ключевыми особенностями метода XGBoost являются:

1. Параллелизация, то есть модель обучается на нескольких ядрах CPU
2. Регуляризация, которая позволяет избежать переобучения
3. Нелинейность в силу того, что «слабым учеником» является дерево решений – XGBoost умеет определять закономерности в данных, даже если они не являются линейными
4. Кросс-валидация, которая встроена внутрь алгоритма

Рассмотрим принцип работы алгоритма XGBoost. В качестве функции ошибки на каждой итерации XGBoost использует функцию, которая минимизирует общую ошибку:

Где – функция потерь,

- i-ый элемент обучающей выборки и сумма предсказаний первых t деревьев,

- признаки i-го элемента,

- дерево решений (функция),

- предсказание модели на i-м элементе,

- регуляризация функции:

(T - кол-во вершин дерева,

– значения весов в листьях,

, - параметры регуляризации)

Первая часть формулы есть функция потерь, которая вычисляет отклонения предсказанных значений от истинных значений в каждом дочернем узле/листе. Вторая часть формулы включает в себя параметр регуляризации , который помогают снизить невосприимчивость модели к отдельным элементам, задаваемый параметр регуляризации , отвечающий за отсечение листьев, а представляет собой значение весов листьев, которое также можно интерпретировать как результат листа. T есть количество вершин, то есть конечных узлов или листьев дерева.

Необходимо минимизировать ошибку модели на обучающей выборке, то есть для каждого t найти минимум функции оптимизации для градиентного бустинга .

Во время первого обучения дерева решений каждое наблюдение имеет равный вес. Затем вес для наблюдений, для которых ошибка больше, увеличивается, и наоборот - вес уменьшается для данных, где ошибка меньше. После данного шага образуется ансамбль из двух полученных моделей, для которого считаются ошибка, и алгоритм повторяется. Последующие деревья есть взвешенные прогнозы, построенные на основе предыдущих моделей.

Параметром остановки дерева является максимальная глубина, затем алгоритм начинает отсечение ветвей дерева в обратную сторону, и такой подход улучшает вычислительную производительность.

Важно отметить, что хоть алгоритм XGBoost и не исследует все возможные структуры деревьев, но строит деревья «жадным» способом, то есть в глубину/по уровням.

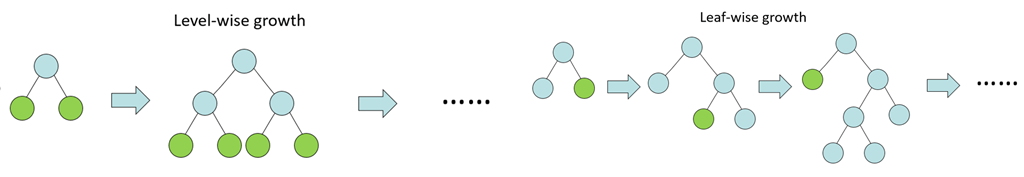


Рисунок 4. XGBoost - разбиение по уровням/в глубину

В свою очередь, LightGBM также является разновидностью методов градиентного бустинга, но, по сравнению с XGBoost, считается более быстрым и производительным и занимает меньше памяти. Это один из наиболее быстрых алгоритмов для обучения и для предсказания.

Он отличается от других моделей градиентной бустинга тем, что использует одностороннюю выборку на основе градиента (GOSS - Gradient Based One Side Sampling) – метод генерации выборки, уменьшающий выборки на основе градиентов. Так как элементы с маленьким градиентом дают меньшую ошибку при обучении, а элементы с большим градиентом дают большую ошибку, последний тип хуже репрезентируется при обучении. Можно избавиться от случаев с маленьким градиентом и обучать только на элементах с большим градиентом, но такой подход сместит распределение выборки. По своей сути, метод GOSS фокусируется на случаях с большим градиентом, но случайно отбирает из генеральной совокупности также и элементы, дающие небольшой градиент.

В архитектуре LightGBM следует принципу разбиения по листьям, вместо разбиения по уровням (в глубину), как XGBoost, и это позволяет алгоритму уменьшить больше потерь, а также является более быстрым способом разбивки. Алгоритм выбирает листья с максимальной функцией потерь и последовательно строит дерево только в этом направлении. Такой метод в результате дает меньшие значения функции потерь, чем рост в глубину, однако рост по листьям также может увеличить сложность модели и привести к переобучению на небольших наборах данных.

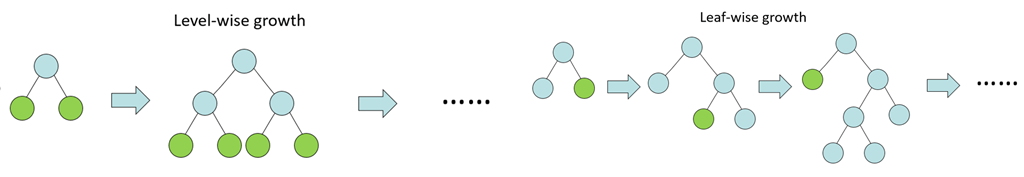


Рисунок 5. LightGBM - Разбиение по листьям

Далее в работе рассмотренные выше ансамблевые методы деревьев решений будут применены при решении прикладных задач, чтобы исследовать каждый из методов и сравнить их эффективность в зависимости от решаемой задачи и рассматриваемых метрик.

Глава 2. Прикладные задачи для анализа и метрики оценки результатов

Исследование методов машинного обучения, в частности, их эффективности, значительно зависит от поставленной задачи и набора данных. Как правило, деревья решений показывают наиболее точные результаты при работе с задачами классификации, так как по своей сути данный алгоритм на выходе предоставляет класс целевой переменной.

Ансамблевые методы, даже учитывая специфику алгоритма дерева решений, могут применяться не только для задачи классификации, но и для задачи регрессии. В случае регрессии, однако, такие методы могут уступать другим алгоритмам обучения (SVM, линейная или полиномиальная регрессия), но при определенном наборе данных, включающим в себя большее количество категориальных признаков, нежели непрерывных, может показывать и лучшие результаты.

Также необходимо учитывать, что при решении задачи регрессии деревья решений не являются хорошим алгоритмом, если требуется предсказать значения, находящиеся вне целевых значений тренировочного датасета, поэтому необходимо применять их только в том случае, если значения целевой переменной находятся в заданном промежутке и не могут выходить за него. Нужно помнить, что ансамблевые модели бустинга, в особенности, адаптивный бустинг, крайне чувствительны к шумам и выбросам в данных, соответственно, при обработке данных для обучающей выборки лучшей практикой будет избавиться от выбросов.

Проводимое исследование ансамблевых методов деревьев решений и их сравнительный анализ будут неполными, если применять их только на задаче классификации или только в задаче регрессии. Поэтому было принято решение использовать два набора данных для более объективного анализа каждого из ансамблевых методов.

Конечно, стоит помнить, что каждая задача регрессии или классификации индивидуальна, и, в зависимости от множества факторов: специфики данных, закономерности в данных, набора признаков, целевой переменной и т.д. ансамблевые модели могут показывать довольно разные результаты. Нельзя сказать, что одна ансамблевая модель подходит лучше для той или иной задачи, так как для каждого набора данных лучшие результаты будут показывать различные модели.

В данной работе рассматриваются две задачи – регрессии и классификации, на каждой из которых проводится сравнительный анализ ансамблевых методов деревьев решений, однако этот анализ будет релевантным для исследуемых случаев и для случаев, где набор данных и задача схожи с рассмотренными.

Два рассматриваемых набора данных относятся к сфере кибербезопасности, являющейся одной из важнейших и развивающейся в большинстве компаний. На данный момент данные организаций уязвимы более, чем когда-либо. Кибербезопасность необходима для качественного управления бизнесом и защиты данных и систем компании, но также является важнейшим инструментом защиты личных данных клиентов. Развитие данного направления значит больше, чем просто использование передовых технологий – в первую очередь, это предотвращение рисков и внештатных ситуаций, которые могут повлечь за собой серьезные и критические последствия для организации.

Модель информационной безопасности в компании построена из множества составляющих: фильтрация писем в корпоративной почте, обучение сотрудников, фильтрация веб-страниц, мониторинг безопасности сети, антивирусы, обновление систем при появлении уязвимостей, файерволы, отдельные системы фильтрации данных по направлению деятельности организации.

К сожалению, на данный момент ни одна модель информационной безопасности компании не является идеальной – всегда присутствует риск ситуации, где системы или данные организации подвергаются атаке в связи с некой уязвимостью в одной или нескольких составляющих модели кибербезопасности. Одной из таких ситуаций может быть утечка персональных данных клиентов, что есть одно из самых критичных нарушений безопасности данных.

При должном уровне внимания организации к направлению кибербезопасности компания может обеспечить себе защиту от больших потерь, как данных, так и активов, и репутации. Помимо защиты данных важно также не снижать приоритетность сопровождения систем информационной безопасности внутри организации.

В работе представлены прикладные задачи из сферы кибербезопасности, в решении которых возможно использование методов машинного обучения, в частности, ансамблевых методов деревьев решений. Первая задача – задача регрессии фокусируется на предсказании суммы потерь компании в случае утечки данных. Таким образом, ансамблевые методы позволяют найти наиболее значимые факторы в определении потерь и оценить риски компании при данной ситуации. Вторая задача классификации относится к более узкой сфере банковской безопасности и исследует систему выявления мошеннических транзакций, которую разрабатывает и сопровождает банк. Ансамблевые методы хорошо подходят для такого рода классификации, так как такие системы опираются на определенный набор признаков, который хорошо интерпретируется и подходит для алгоритма дерева решений.

2.1 Прикладная задача регрессии

Как уже было упомянуто выше – задача регрессии, то есть предсказание числового значения целевой переменной, представляет собой прогнозирование суммы потерь компании в случае утечки данных. Построение модели, которая успешно предсказывает сумму убытков компании, в дальнейшем может помочь спрогнозировать эту сумму в случае внештатной ситуации, основываясь на признаках и причинах возникновения утечки. Также, исследовав набор данных и модель, можно выяснить, какие именно атрибуты влияют на рост суммы потерь, что поможет взять во внимание и предусмотреть все риски.

Набор данных для этой задачи был взят из открытого источника - он представляет набор элементов/список случаев утечки информации в компаниях. Каждый случай утечки данных имеет набор атрибутов, характеризующий этот конкретный случай. Целевой переменной в этом датасете является непосредственно сумма потерь компании в отдельном случае утечки данных.

Набор данных не является большим (2000 строк и 9 столбцов), что снизит эффективность обучения модели. Однако в ансамблевых методах есть элементы, которые могут помочь бороться с этой проблемой.

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6. Срез данных задачи регрессии

Среди столбцов датасета есть те, что содержат уникальные значения и не будут считаться признаками в обучающей и тестовых выборках: Название компании и Описание, которое содержит подробное описание произошедшего. Будущими же атрибутами модели могут стать следующие категориальные столбцы: Штат, Тип бизнеса компании, Тип утечки, Местонахождение утерянных данных, Участие сотрудника компании, а также один числовой столбец с данными о Количестве пострадавших от утечки данных людей. Стоит отметить, что под утечкой данных в этой задаче подразумевается утечка персональных данных клиентов организации.

Количество людей, которые пострадали от утечки данных, расположено на интервале [0;80 000 000]. Если расположить значения целевой переменной на графике, то будет очевидно, что в данных присутствуют выбросы. Большая часть значений признака располагаются на промежутке, намного меньшим 10 000 000.

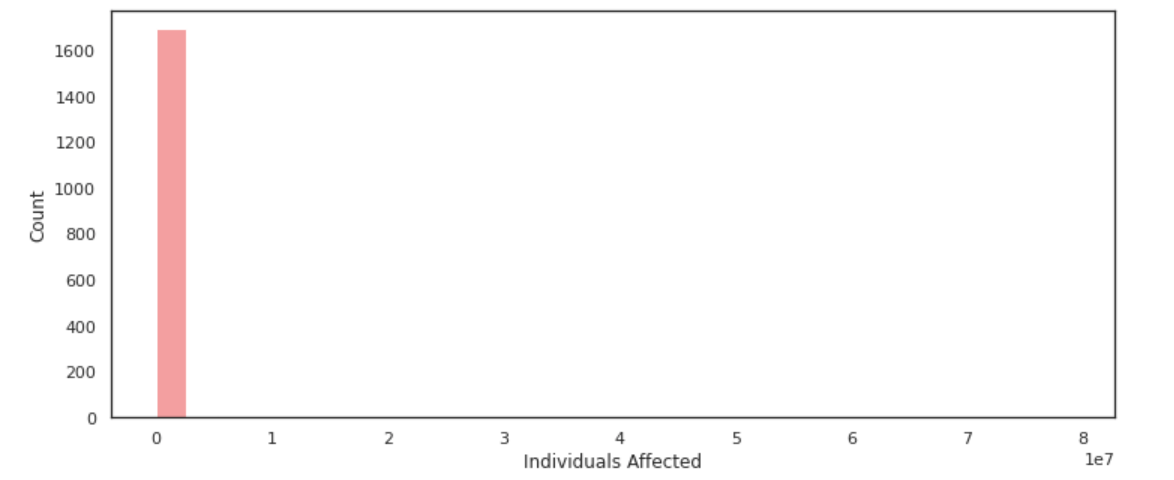


Рисунок 7. Количество пострадавших людей (с выбросами)

Так как выбросы будут плохо влиять на качество модели, в особенности, на ансамблевые модели деревьев решений, необходимо избавиться от выбросов. Рассмотрев визуализацию распределения значений, можно сказать, что большая часть случаев утечки данных затронула небольшое количество людей в пределах 40 000 человек. В исследуемом датасете утечка данных чаще всего имела не массовые последствия, а индивидуальные.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 8. Количество пострадавших людей

Целевой переменной является столбец с потерями компании, значения которого располагаются от 1000 до 1 миллиона долларов. После исключения выбросов в столбце «Количество пострадавших» максимальное значение потерь также сократилось до 300 000 долларов. На графике ниже представлено распределение значений целевой переменной:

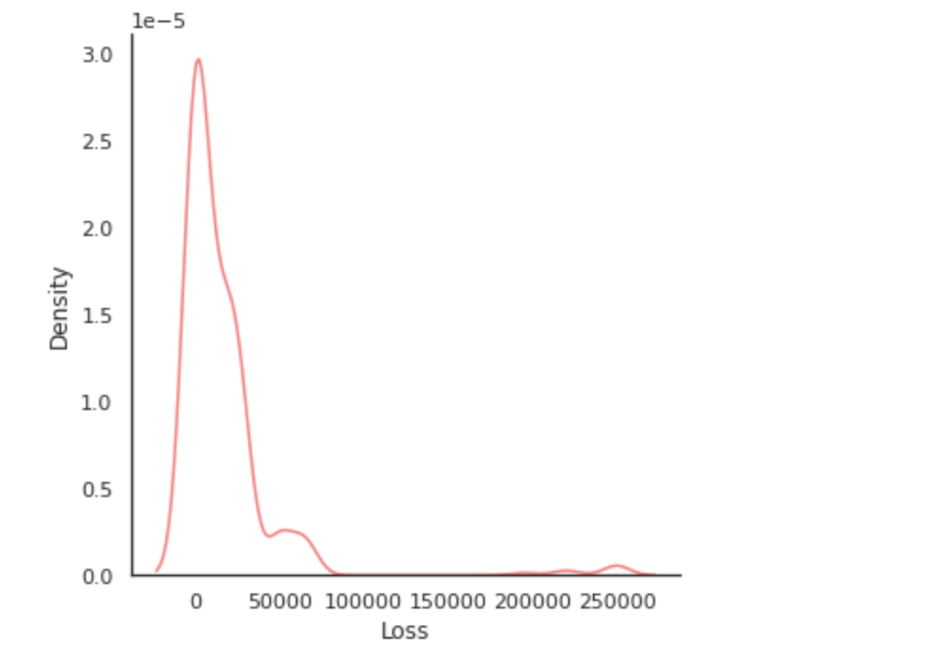


Рисунок 9. Потери компании в случае утечки данных

Можно заметить, что и в этом случае потери компании смещены к наименьшему значению, что допускает существование корреляции с признаком «Количество пострадавших». Тем не менее, значение корреляции между данным атрибутом и целевой переменной находится на уровне 0.3, то есть зависимость целевой переменной от количества пострадавших человек не строго линейна. Также присутствует корреляция в 0.15 между целевой переменной и годом случая утечки данных, что может говорить о том, что с течением времени данные организации становятся более дорогим «имуществом», а потери компании в случае их утечки, а также другие последствия таких случаев растут.

Изображение выглядит как квадрат

Автоматически созданное описание

Рисунок 10. Корреляция числовых признаков и целевой переменной

Категориальные признаки, которые предоставляют наиболее ценную информацию, относящуюся непосредственно к факторам и причинам утечки данных, это столбцы «Тип утечки» и «Местонахождение утерянной информации».

Первый признак состоит из 28 уникальных значений, среди которых кража, утеря, хакерский инцидент, несоответствующее удаление, неразрешенный доступ или раскрытие и т.д. С помощью этого атрибута можно определить основные и наиболее распространенные причины утечки информации: чаще всего причиной утечки данных является кража: ~42% от всех причин утечки, далее следует неразрешенный доступ или раскрытие информации ~23%, и 3-м наиболее распространенным типом утечки в ~7,5% является потеря данных.

Атрибут «Местонахождение утерянной информации» содержит 60 уникальных значений, из которых настольный компьютер, ноутбук, e-mail, сетевой сервер, бумага и т.д. Так как утерянные данные могут находиться на нескольких источниках сразу, то в данном столбце некоторые категории могут быть совмещены.

Признаком с наименьшим количеством категорий является «Участие сотрудника компании», которое включает в себя лишь два значения: Да или Нет.

Таким образом, задача регрессии будет заключаться в предсказании суммы потерь компании в случае утечки данных. Она будет решаться с помощью ансамблевых моделей, используя набор категориальных и численных признаков, каждый из которых характеризует один отдельный случай утечки данных в организации. Метрики оценки результатов работы моделей, в том числе, точности предсказаний для этой задачи будут описаны в третьем разделе текущей Главы 2.

2.2 Прикладная задача классификация

Задача типа классификации также является актуальной в области кибербезопасности, но уже в более узкой банковской сфере. Банковские организации каждую минуту пропускают через свои системы огромное количество транзакций между счетами. Каждая транзакция должна быть проверена на предмет мошенничества, и в случае обнаружения такого вида транзакции, немедленно быть прервана.

В банковских организациях существуют специальные системы, которые позволяют анализировать проходящие через банковские системы транзакции и проверять их на наличие мошенничества (фрода). В крупных банках такие системы разрабатываются внутри организации и сопровождаются отдельным направлением, чаще всего, в департаменте кибербезопасности. Такие системы должны работать непрерывно, и ее отключение может привести к серьезным последствиям и пропуску ряда фродовых транзакций. В организациях зачастую существуют и резервные системы на случай отключения основной, а сопровождение основной и резервной систем являются важной составляющей обеспечения кибербезопасности в организации.

Такие системы опираются на анализ совокупности признаков проходящей через систему транзакции. Алгоритм проверяет каждый из признаков входящей транзакции и выносит решение о прерывании или пропуске транзакции. Примерами таких анализируемых признаков могут быть: дата и время транзакции, отправитель и получатель, место проведения, история транзакций получателя и отправителя, переводимая сумма и т.д. Признаки имеют разные веса и приоритетность и играют разную роль в определении типа транзакции.

Деревья решений, а точнее ансамблевые методы, могут быть довольно эффективными методами в решение такого вида прикладной задачи, так как имеется набор (зачастую категориальных) признаков, которые разделяют мошеннические и не-мошеннические транзакции на два класса. Большинство компаний, которые занимаются финансовыми транзакциями, для выявления мошеннических транзакций, используют как раз модели, основывающиеся на деревьях решений, такие как XGBoost, но хорошей практикой является и рассмотрение других моделей.

Итак, необходимая задача классификации есть определение по набору входящих численных и категориальных признаков, является ли транзакция обычной, или же в систему попала фродовая/мошенническая транзакция, которую необходимо прервать. По своей сути, это есть задача классификации, которая решается с помощью дерева решения классификации.

Классификация подразумевает под собой предсказание класса целевой переменной. В рассматриваемой задаче целевой переменной является столбец с присвоенным транзакции классу - мошенническая (1) и не мошенническая (0).

Важно отметить, что в силу специфики задачи, важна не только верная классификация транзакции, но скорее минимизация доли неверно классифицированных фродовых транзакций – алгоритм не должен допустить того, чтобы мошенническая транзакция прошла через систему и завершилась. Кроме того, неверная классификация не мошеннических транзакций представляется возможной и несет за собой меньше последствий.

Таким образом, главной целью модели является верная классификация всех фродовых транзакций (минимальное значение метрики False Negative и максимальное значение метрики True Positive), что может повести за собой увеличение количества неверно классифицированных обычных транзакций (False Positive). По своей сути, необходимо максимизировать метрику recall. По этой причине для исследования эффективности модели будет использована метрики F-1 и AUPRC, которые будут рассмотрены в третьем разделе текущей главы.

Используемый для решения задачи классификации датасет включает в себя 284807 строк и 31 столбец. Большая часть столбцом являются непосредственно трансформированными признаками транзакции V1-V28. В наборе данных также присутствуют не трансформированные признаки: Время, Сумма и Класс, последний из которых является целевой переменной.

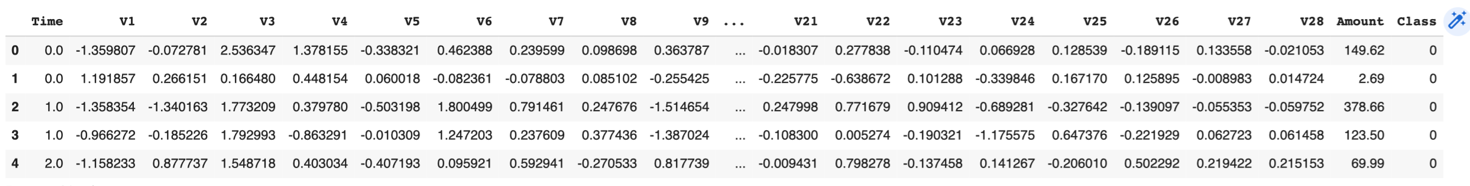


Рисунок 11. Срез данных задачи классификации

В силу конфиденциальности антифродовой системы банка, которому принадлежит используемый датасет, признаки от 1 до 28 были обезличены, поэтому сделать первоначальный анализ, основываясь на категориях признаков, нет возможности. Сами значения атрибутов есть результат трансформации на основе метода главных компонент, проведенного, по всей видимости, для уменьшения размерности данных. Даже опуская исследование содержимого атрибутов, с помощью исследования корреляции между признаками можно определить, есть ли в наборе данных признаки, которые сильно коррелируют между собой и могут повлиять на модель.

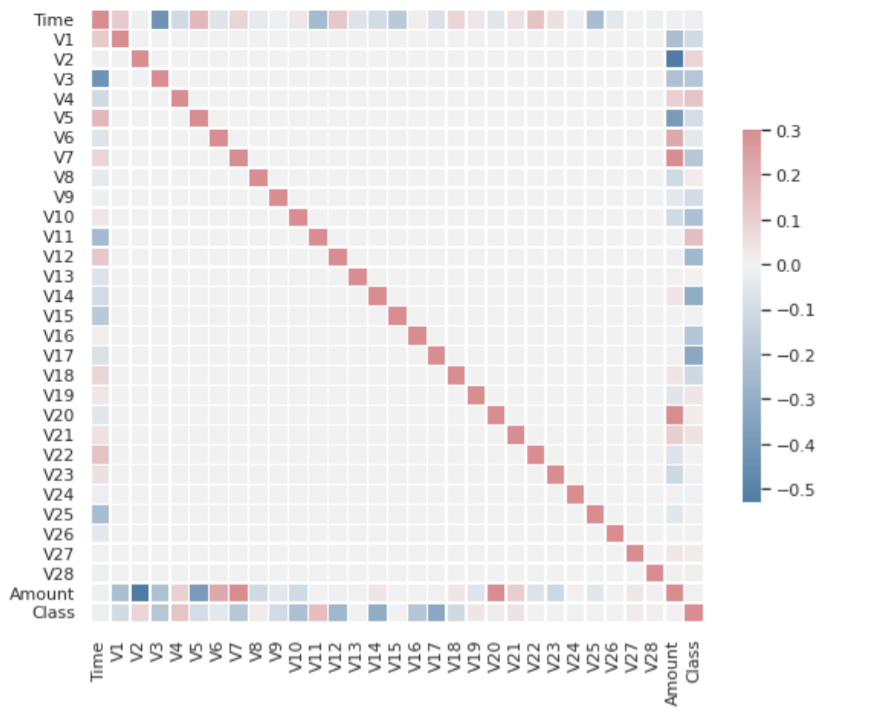


Рисунок 12. Матрица корреляции

По тепловой карте матрицы корреляции можно заметить, что корреляции между самими признаками V1-V28 не наблюдается. Существует отрицательная корреляция между атрибутом «Сумма» и признаком V2 со значением в -0.5, а также корреляция этого же столбца и признаком V7 со значением 0.3. Однако так как невозможно определить, какую информацию несут в себе признаки, а также в силу того, что сумма является атрибутом, по своему смыслу и содержанию отличному от признаков V2 и V7, к тому же, с не самыми большими значениями коэффициента корреляции, то не представляется возможным объединить эти признаки в один, чтобы избавиться от корреляции между столбцами набора данных, следовательно, от риска мультиколлинеарности. В свою очередь переменная класса имеет отрицательную положительную корреляцию с некоторыми компонентами V1-V28, но не имеет такого типа зависимости от переменных «Время» или «Сумма».

Рассмотрим два признака, доступные для интерпретации: Время и Сумма. Первый атрибут представляет собой количество прошедшего времени между каждой транзакцией и первой транзакцией набора данных.

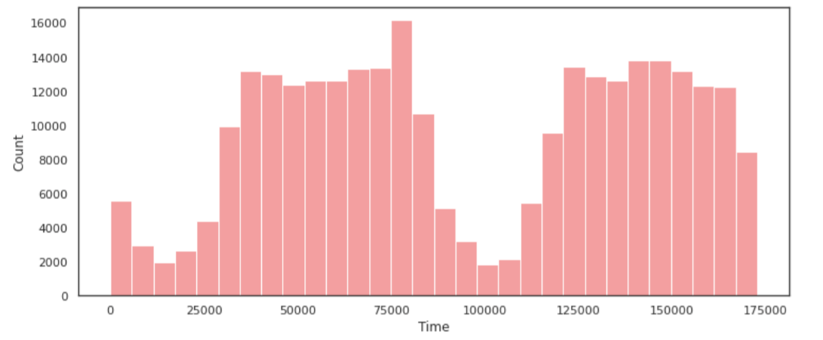


Рисунок 13. Распределение значений времени для всего датасета

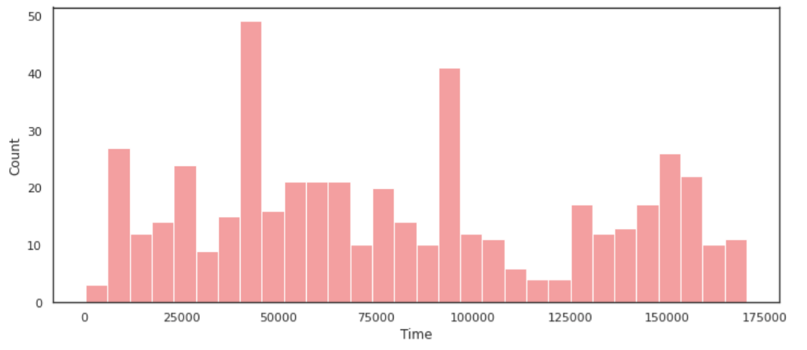


Рисунок 14. Распределение значений времени для мошеннических транзакций

Можно сказать, что в этом признаке нет четкого индикатора того, какой промежуток времени принимает на себя большее количество мошеннических транзакций. Меньше всего транзакций были обработаны в промежутке в окрестностях значения 1000000. При этом в этом же промежутке было обнаружено фродовых транзакций второго по количеству значений. Однако из-за небольшого числа значений этого класса, около 40 мошеннических транзакций, обнаруженных в этот момент, не являются большой долей от всего 2000 общего числа, хотя такая закономерность тоже имеет место быть.

Второй признак «Сумма» показывает сумму транзакции в денежных единицах.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15. Распределение суммы по всему датасету

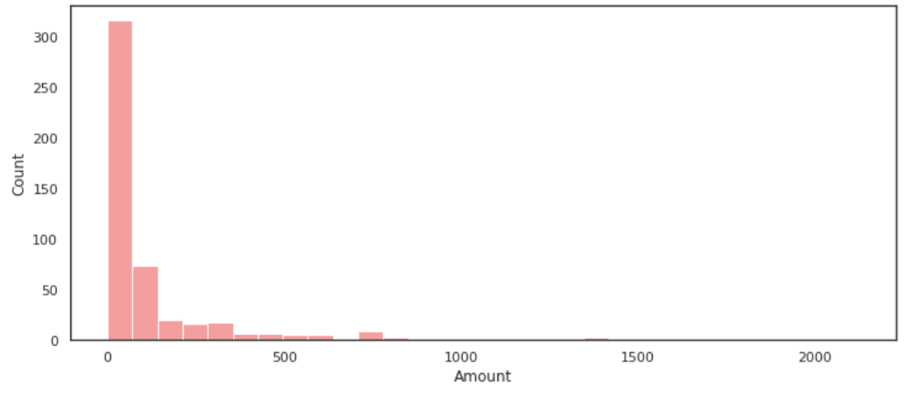


Рисунок 12. Распределение суммы по мошенническим транзакциям

Множество значений данного признака имеет выбросы, большая концентрация значений находится в промежутке от 0 до 1000. Рассмотрев распределение значений суммы транзакций в срезе мошеннических, можно сказать, что выбросы в полной выборке относились к не фродовым транзакциям. В основном, сумма мошеннических транзакций лежит в промежутке от 0 до 100, но в этом же промежутке находится большая часть суммы всех транзакций, так что хоть этот признак и дает некую информацию о закономерности, очевидной зависимости класса от суммы проводимой транзакции не наблюдается.

Главная особенность исследуемой задачи является в несбалансированности используемых для обучения алгоритма данных – через такие системы финансовых организаций проходит огромное количество транзакций, из которых мошенническими является очень маленькая часть. В рассматриваемом наборе данных мошеннические транзакции также составляют очень маленькую долю от всех транзакций 492 фродовые транзакции на общее количество в 284807, то есть в соотношении это 492 мошеннические к 284315 не мошенническим. Исследуемый датасет является крайне несбалансированным, где транзакции класса 1 (фродовые) составляют менее 1% от общего количества транзакций, а именно всего 0,17% даже при довольно большом наборе данных.

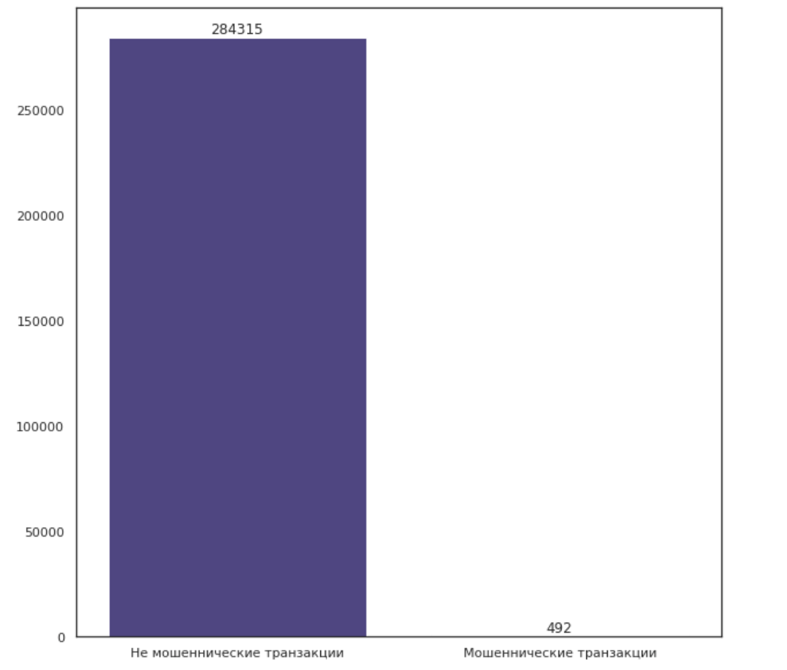


Рисунок 13. Распределение значений целевой переменной по классам

Такая особенность может повлиять на оценку результатов работы модели. Если использовать исходный набор данных как базу для предсказывающей модели, можно получить большое число ошибок, а также столкнуться с проблемой переобучения, так как алгоритмы будут считать, что подавляющее большинство транзакций мошенническими не являются. От модели требуется не приходить к таким выводам, а обнаружить закономерности в признаках, которые сигнализируют о том, что входящая транзакция является фродом.

Как уже было упомянуто выше, необходимо максимизировать метрику recall, не забывая о желательно низком уровне метрики неверно классифицируемых обычных транзакций. Однако при использовании такого несбалансированного датасета данная False Positive Rate может оставаться низкой даже при классификации большого числа обычных транзакций как мошеннических, потому что в результате будет все еще большое число верно определенных не мошеннических транзакций.

Искусственно сбалансировать датасет не представляется возможным, так как будет потерян важный фактор, присутствующий также и в реальных данных. Если обучать модель на сбалансированных данных, то это поможет достичь лучших результатов на валидационной выборке, однако целью является оптимизировать результат на несбалансированных данных. Для того, чтобы решить эту проблему, для обучения можно использовать все мошеннические транзакции, но из не-мошеннических транзакций выбрать лишь срез/подвыборку.

Кроме того, в случае использования методов бэггинга и случайного леса возможны наилучшие результаты, так как применение метода формирования выборок для базовых алгоритмов бутстрэпинг может помочь преодолеть несбалансированность датасета.

2.3. Метрики оценки результатов

Для определения точности предсказаний в каждом типе задач будут использоваться наиболее подходящие метрики оценки результатов работы модели, которые учитывают специфику рассматриваемой прикладной задачи – эти метрики будут рассмотрены подробнее ниже.

Кроме оценки точности предсказаний для каждой ансамблевой модели будет вычисляться время исполнения модели, чтобы определить, какие ансамблевые модели являются наиболее эффективными по времени исполнения – характеристика, которая является важной практически в любом виде задач, решающих реальные проблемы на реальных данных.

Если модель показывает наилучшую точность, но ее исполнение занимает много времени, или же наоборот – модель имеет меньшую точность, но обучается и работает быстрее, то, прежде чем применять ту или иную модель, необходимо учесть все риски и понять, какая из метрик эффективности является более приоритетной в той или иной ситуации.

Итак, ниже будут рассмотрены метрики точности предсказаний для рассмотренных в первом и втором разделах текущей главы задач.

Как уже было упомянуто, деревья решений обычно не являются наиболее эффективным методом для решения задач регрессии, однако при использовании ансамблевых моделей появляется возможность получить достаточно точные результаты. Оценкой точности предсказания в задаче регрессии была выбрана наиболее распространенная метрика при рассмотрении регрессии - среднеквадратическая ошибка:

,

Где N – количество элементов набора данных,

– истинные значения целевой переменной,

– предсказанные значения целевой переменной.

Оценка же точности предсказаний модели в случае рассматриваемой задачи классификации может быть довольно сложной задачей. Выше была приведена цель работы построенной модели – достичь максимального значения True Positive Rate, сохраняя минимальное значение False Negative Rate, то есть добиться максимального значения метрики recall.

Другие метрики оценки точности предсказаний в классификации будут нерепрезантивны из-за несбалансированности набора данных. Значение, например, самой распространенной метрики accuracy всегда будет высоким и близким к 1, так как количество верно классифицируемых обычных транзакций всегда будет большим по сравнению с любыми классифицируемыми мошенническими транзакциями.

Сама метрика recall также не даст сделать объективные выводы при несбалансированности датасета, поэтому для оценки точности предсказаний будет использована метрика F-beta, включающая в себя равнозначные  или весовые вклады значений метрик precision и recall и являющаяся их гармоническим средним:

  Коэффициент β отвечает за вклад отдельной метрики precision или recall в значение F-метрики. Наиболее распространенной является метрика F-1, допускающая равнозначный вклад обоих значений, однако в случае рассматриваемой несбалансированности будет важнее вклад метрики recall, так она учитывает значения False Negative – в данном случае, это те мошеннические транзакции, которые ошибочно были классифицированы как не мошеннические, что является как раз тем значением, которое модели необходимо минимизировать, то есть максимизировать recall.

По этой причине в сравнении точности предсказаний будет использована метрика F-2, где recall имеет больший вклад в значение метрики:

Вторая используемая для вычисления точности предсказания метрика – это метрика AUPRC (area under the precision-recall curve), площадь под кривой precision-recall. Данная метрика также использует метрики precision и recall и применяется в случае несбалансированности датасета, когда важно нахождение элементов, принадлежащих к классу 1. Если модель достигает идеального значения AUPRC, значит, она верно классифицировала все значения, принадлежащие к классу 1 (мошеннические транзакции), то есть достигла максимального значения метрики recall, а также не произвела неверных классификаций (обычные транзакции как мошеннические).

Базовым значением метрики AUPRC, в отличие от других метрик точности классификации, является уникальным значением для каждого набора данных и равняется отношению:

,

Где – количество мошеннических транзакций, а – количество обычных транзакций.

То есть для рассматриваемой задачи базовым значением этой метрики будет 0,0017, поэтому полученная метрика AUPRC будет оцениваться в отношении к базовой вычисленной метрике.

Опираясь на две описанные выше метрики для задачи классификации, можно оценить, насколько точно были классифицированы все фродовые транзакции. Стоит отметить, что ни в одной из метрик не используется значение True Negatives, потому что из-за несбалансированности датасета и большого количества обычных не мошеннических транзакций, это значение всегда будет большим и не даст объективной оценки точности предсказаний модели.

Определив набор метрик для оценки точности предсказаний для каждого типа задач, а также общую метрику времени работы модели, можно перейти к непосредственному построению и сравнению ансамблевых моделей деревьев решений.

Глава 3. Построение моделей и сравнение ансамблевых методов

3.1 Исследуемые ансамблевые модели и их параметры

Сравнительный анализ ансамблевых методов деревьев решений, а именно бэггинга, случайного леса и бустинга, проводится на двух типах задач и двух наборах данных. Исследование эффективности работы каждой из ансамблевых моделей поможет определить, какие модели лучше справляются с задачей регрессии, а какие с задачей классификации. Таким образом, можно сказать, что сравнительный анализ в данной работе происходит на двух уровнях: на уровне решаемой задачи (сравнение результатов моделей в конкретной прикладной задаче) и на уровне типов задач (сравнение результатов работы моделей в зависимости от решаемой задачи).

По этой причине в решении обеих прикладных задач были исследованы одинаковые алгоритмы. В Главе I были перечислены и описаны методы, которые будут использованы в работе, а также соответствующие им применяемые модели.

Для каждой модели была реализована кросс-валидация. Каждая модель после обучения и проверки на валидационной выборке производит набор метрик оценки эффективности, по которым в конце проводится сравнительный анализ рассматриваемых ансамблевых моделей.

Параметры для каждой модели подбирались с помощью модуля GridSearchCV, то есть были выбраны параметры, с которыми модель давала наилучший результат. GridSearchCV находит лучшие параметры, перебирая все комбинации параметров и создавая модель для каждой из них. Такой подбор может занимать много времени в зависимости от сложности и размера модели, а также, в зависимости от размера используемого набора данных. Существуют другие практики подбора параметров, однако в данном случае необходимо повысить эффективность работы непосредственно ансамблевого метода деревьев решений.

Из методов бэггинга будут рассмотрены непосредственно бэггинг и случайный лес. Методы бэггинга подразумевают построение нескольких независимых моделей деревьев решений, элементы для которых формируются из исходного набора данных методом бутстрэпинга, то есть с заменой. Случайный лес есть модификация алгоритма бэггинга, в котором базовые алгоритмы деревьев декоррелируются еще больше за счет формирования подвыборки признаков набора данных.

Модель базового бэггинга представлена регрессором sklearn.ensemble.BaggingRegressor и классификатором sklearn.ensemble.BaggingClassifier, оба принадлежат библиотеке sci-kit learn.

Основными параметрами моделей бэггинга, отвечающих за особенности метода, являются: base\_estimator - базовый ученик, что в нашем случае дерево решений регрессии или классификации, n\_estimators - количество базовых учеников, max\_samples – количество выбираемых из набора данных элементов, max\_features – количество выбираемых из набора данных признаков, bootstrap (True/False) – является ли набор данных для базовых учеников сгенерированным методом бутстрэпинга.

Модели случайного леса также представлены в библиотеке sci-kit learn – это sklearn.ensemble.RandomForestRegressor и sklearn.ensemble.RandomForestClassifier.

Основными параметрами данной модели являются: n\_estimators – количество базовых учеников, то есть деревьев решений, criterion – функция, измеряющая качество («чистоту») сплита, для задач регрессии и классификации будут использованные рассмотренные выше квадратичная ошибка и индекс Джини соответственно, max\_depth – максимальная глубина дерева, min\_samples\_split – минимальное количество элементов, необходимых для разбиения узла, min\_sample\_leaf – минимальное количество элементов в узле, чтобы считать его конечным (листом), max\_features – максимальное количество выбираемых признаков при нахождении наилучшего узла, max\_leaf\_nodes – наибольшее количество листьев, min\_impurity\_decrease – узел будет разбит, если данный сплит способствует уменьшению значения не чистоты узла на величину равную или больше данного параметра, bootstrap - является ли набор данных для базовых учеников сгенерированным методом бутстрэпинга, class\_weight – задание весов для классов, max\_samples – количество элементов выборки, сгенерированной методом бутстрэп для каждого базового ученика.

Рассматриваемыми метода бустинга будут адаптивный бустинг и градиентный бустинг, последний из которых представлен двумя моделями. Ансамблевые методы бустинга последовательно обучают слабых учеников, учитывая ошибки предыдущих. Адаптивный бустинг учитывает отклонения, обновляя значения весов для каждого вхождения набора данных в зависимости от того, насколько оно повлияло на ошибку, а градиентный бустинг на каждом шаге оптимизирует выбранную функцию потерь.

Адаптивный бустинг строится с помощью моделей из библиотеки sci-kit learn sklearn.ensemble.AdaBoostRegressor и sklearn.ensemble.AdaBoostClassifier.

Пользовательскими данных ансамблевых моделей являются: base\_estimator – базовый алгоритм («слабый ученик»), то есть дерево решений, n\_estimators - количество слабых учеников, learning\_rate – коэффициент скорости обучения, то есть вес, применяемый к каждому регрессору или классификатору на каждой итерации бустинга так, что чем выше коэффициент скорости обучения, тем больше вклад отдельного регрессора или классификатора, loss для модели регрессии – функция потерь, использующаяся для обновления весов на каждой итерации.

Градиентный бустинг представлен двумя типами моделей XGBoost и LightGBM. Каждый из этих методов представлен в своих библиотеках соответственно: xgboost и lightgbm.

Основными настраиваемыми модели XGBoost являются max\_depth – максимальная глубина одного дерева, learning\_rate – коэффициент скорости обучения, n\_estimators – количество слабых учеников, max\_leaves – максимальное число листьев, gamma – минимальное значение уменьшения функции потерь, необходимое для последующего разбиения на текущем узле дерева. В качестве же параметров модели LightGBM выступают похожие: boosting\_type – тип бустинга, в этой работе дерево решений градиентного бустинга, num\_leaves – максимальное количество листьев дерева для базовых учеников, max\_depth – максимальная глубина базового алгоритма, learning\_rate – коэффициент скорости обучения, n\_estimators – количество бустинговых деревьев («слабых учеников»), subsample\_for\_bin – количество элементов для создания ячеек, параметр уникальный для LightGBM, который использует оптимизированный алгоритм обучения деревьев решений, основанный на гистограммах, min\_split\_gain - минимальное значение уменьшения функции потерь, необходимое для последующего разбиения на текущем узле дерева и другие параметры, относящиеся к формуле оптимизации функции потерь.

 Итак, рассмотрев все используемые модули библиотек для построения ансамблевых моделей, можно приступить к построению моделей и их сравнительному анализу.

3.2 Построение моделей и получение результатов в задаче регрессии

В Главе II был проведен предварительный анализ данных и подготовка данных к моделям. Столбцы с уникальными значениями не важны для построения модели, поэтому атрибуты «Название компании» и «Описание случившегося» будут убраны из обучающей выборки. То, что тоже необходимо сделать перед построением моделей – привести категориальные признаки к виду, который сможет использовать модель. С помощью LabelEncoder категориальные признаки были приведены к подходящему для включения в модель виду, где каждый уникальный признак закодирован определенным числом.

Таким образом, обучающая и валидационная выборки состоят из набора признаков и целевого столбца, который в данной прикладной задаче представляет собой сумму потерь компании в случае утечки данных.

Для демонстрации работы алгоритма дерева решений построим однокомпонентную модель регрессионного дерева решений DecisionTreeRegressor. Оптимальные параметры для этой модели также были избраны путем перебора всех моделей с возможными параметрами с помощью GridSearchCV.

Изображение выглядит как лодка

Автоматически созданное описание

Рисунок 14. Визуализация части построенного регрессионного дерева решений

В каждом узле дерева решений можно найти информацию об условии разбиения узла на последующие, которое включает рассматриваемый на этом этапе признак, среднеквадратичную ошибку на текущем узле, количество элементов в данном узле и предсказанное значение. Построенная однокомпонентная модель является не самой большой и легко интерпретируемой в силу небольшого размера обучающего набора данных. С помощью ансамблевых методов можно агрегировать результаты нескольких базовых алгоритмов деревьев решений и получить более сложные модели с более точными результатами.

Построив все модели и вычислив среднеквадратичное отклонение для каждой модели, можно сравнить метрики точности предсказания результатов для каждого метода:

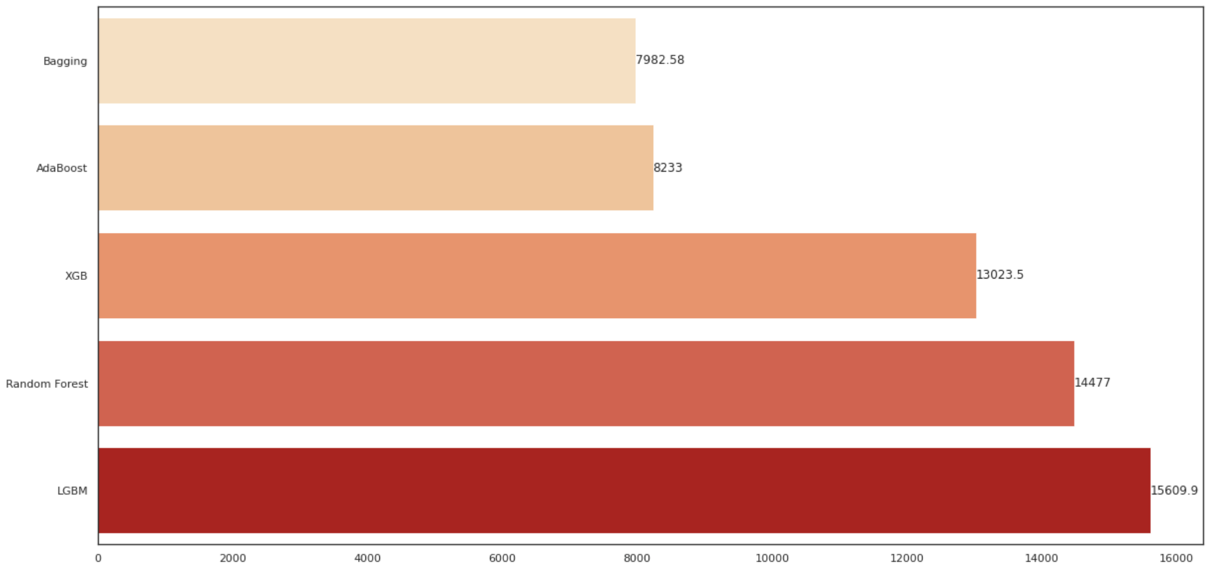


Рисунок 15. Среднеквадратичные ошибки ансамблевых моделей

По метрикам оценки точности предсказаний можно увидеть, что наилучший результат показала модель бэггинга со среднеквадратичным отклонением в 7982.58, которое указывает на среднее отклонение в ~89,3 денежных единиц, что является хорошим результатом, учитывая то, как смещены значения целевой переменной. Хороший результат можно объяснить тем, что метод формирования выборок для базовых алгоритмов бутстрэп помог справиться с проблемой небольшого размера датасета. Кроме того, хоть случайный лес и является разновидностью метода бэггинга, он показал результаты хуже по причине небольшого количества признаков в наборе данных.

Кроме этого, модель адаптивного бустинга также показала хорошую метрику точности предсказаний, значение которой находится довольно близко к результату работы бэггинга. Тем не менее, модели градиентного бустинга хоть и показали приемлемую среднеквадратичную ошибку, уступают в точности первым двум. Наихудшую точность предсказания показала модель LightGBM в среднеквадратичное отклонение равное 15609, то есть средним отклонением предсказанный значений целевой переменной примерно в 124,9.

Однако при сравнении затраченного времени на обучение и валидацию моделей, LightGBM, показавшая наименьшую метрику точности, находится на первом месте с наименьшим затраченным временем в 0.25 с.

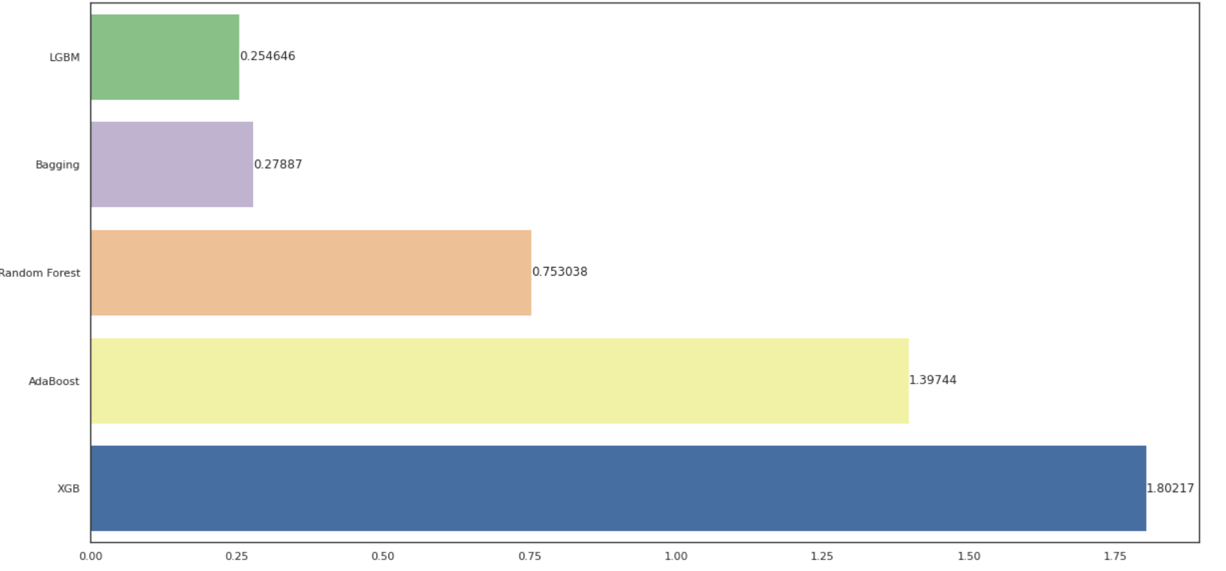


Рисунок 16. Затраченное время ансамблевых моделей регрессии

LightGBM считается наиболее легким и быстрым из ансамблевых методов из-за оптимизированного алгоритма построения деревьев решений, однако при этом уступает другим моделям в точности. На втором месте находится модель бэггинга, чье исполнение заняло 0.28 с. Случайный лес есть модифицированная версия бэггинга, построение которой заняло немного больше времени. Модели адаптивного и градиентного бустинга стали наиболее долгими для реализации. Разрыв между временем исполнения двух методов градиентного бустинга самый большое и составляет 1,55 с, что наглядно показывает, насколько быстро работает LightGBM по сравнению с XGBoost.

Так как метрик оценки несколько, то для того, чтобы выбрать наилучшую модель, необходимо установить, какой из параметров сравнения является более приоритетным в решении данной задачи. Тем не менее, на исследуемых данных лучше всего показала себя модель бэггинга, имея наилучшую точность предсказания целевых значений и второе по скорости исполнения время.

* 1. Построение моделей и получение результатов в задаче классификации

Для рассмотрения моделей задачи классификации необходимо также провести предварительную обработку данных. Все столбцы уже являются числовыми, однако в отличие от нормализованных признаков V1-V28, значения столбцов «Время» и «Сумма» не обработаны. Различие в масштабах данных может негативно повлиять на результаты работы модели. Для стандартизации данных столбцов применим функцию StandardScaler(), которая стандартизирует функции в вид со средним значением 0 и стандартным отклонением в 1.

После приведения данных к нужному виду, можно проверить, как выглядит однокомпонентная модель дерева решений классификации. В узлах также находится информация об условии разбиения узла на последующие, которое включает рассматриваемый на этом этапе атрибут, значение индекса чистоты сплита, количество элементов в данном узле и предсказанный класс.

Изображение выглядит как карта, линия, несколько

Автоматически созданное описание

Рисунок 17. Визуализация части построенного дерева решений классификации

В рассматриваемой задаче классификации метрики точности были выбраны согласно специфике датасета и цели работы моделей. Приоритетом является нахождение всех мошеннических транзакций, то есть максимальное значение True Positive и минимальное значение False Negative. Соответствие этому измерению точности предсказаний представляет собой метрика recall. Для того, чтобы оценить результаты работы модели на несбалансированном наборе данных, в качестве метрик были выбраны две метрики, которые показывают состояние метрики recall – f2-score и AUPRC. Рассмотрим их ниже.

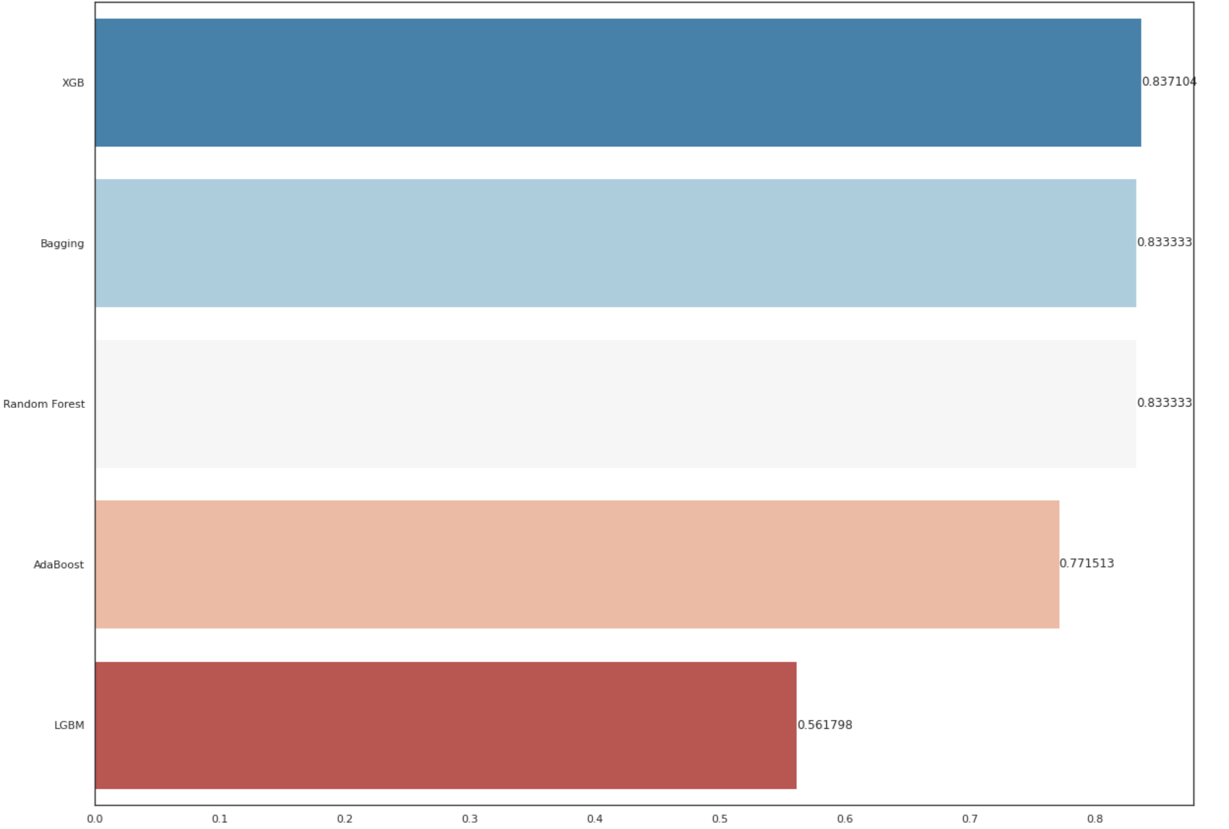


Рисунок 18. Метрики F-2 ансамблевых моделей

Наилучшие значения метрики f-2 показали модели XGBoost, Бэггинг и Случайный лес, значение которых превысило 0.83. Это значит, что эти модели имеют высокое значение верно классифицированных мошеннических транзакций. Другие модели, в частности, адаптивный бустинг находится на отметке 0.77, а значит, что алгоритм допускает неверную классификацию фродовых транзакций как обычных. Наконец, модель LightGBM имеет метрику f-2 с наименьшим значением в 0.56 – данная модель плохо выявляет мошеннические транзакции.

Рассмотрим метрику AUPRC, чтобы выяснить, является ли эффективность моделей, оцененная по метрике f-2 такой же, как и при рассмотрении метрики AUPRC.

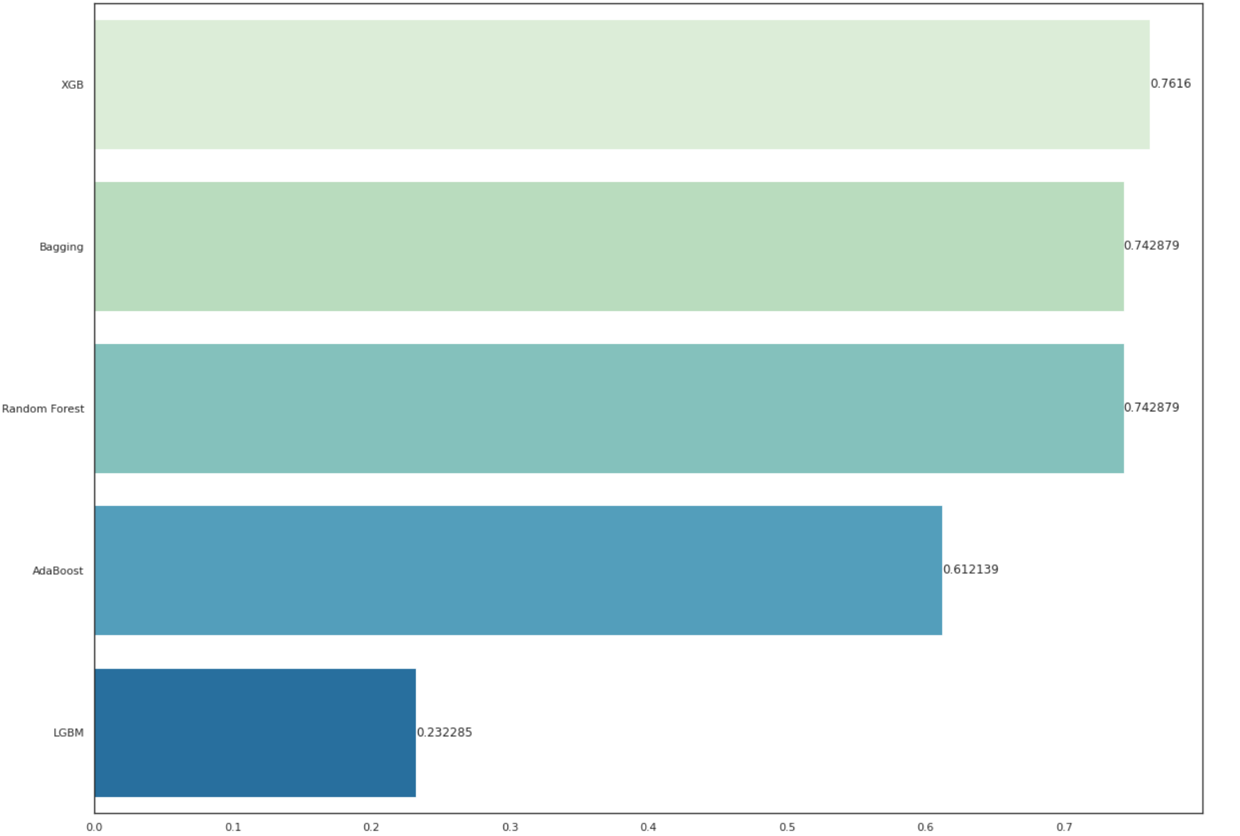


Рисунок 19. Метрики AUPRC ансамблевых моделей

Особенность метрики AUPRC в том, что базовым значением является доля предсказываемого положительного класса от общего количества элементов набора данных, в исследуемой задаче таким значением является 0.0017. Пусть вычисление метрике AUPRC отличается от вычисления метрики F-2, порядок моделей, обусловленный точностью предсказаний, является одинаковым. По этой причине можно сделать вывод, что вне зависимости от используемой метрики точности предсказаний, лучшие результаты показали модели XGBoost и модели бэггинга, а худшее значение модель LightGBM.

Необходимо также проверить, насколько быстро работают построенные модели. Учитывая специфику задачи, то есть обработку транзакций, которая должна происходить как можно быстрее, при этом не теряя в точности предсказаний, метрика затраченного времени является существенной для определения общей эффективности.

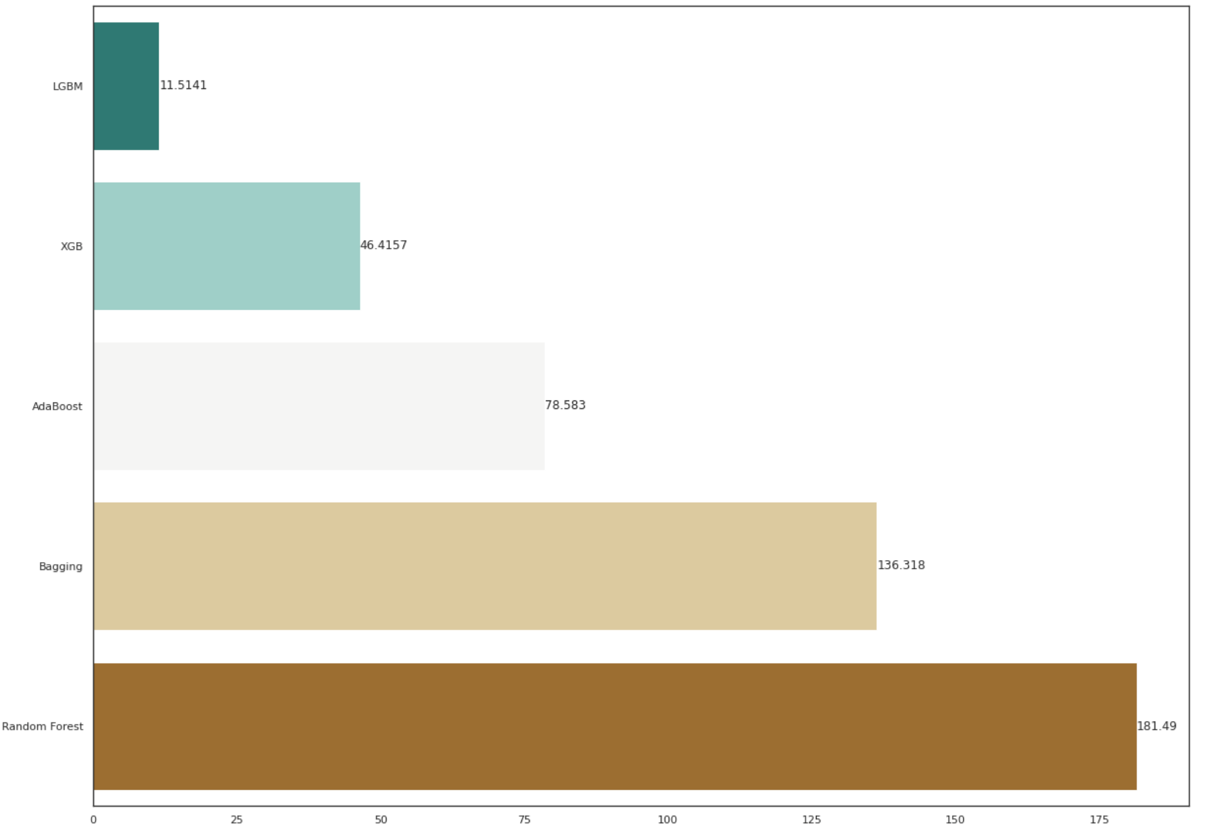


Рисунок 20. Затраченное время ансамблевых моделей классификации

Разница между временем работы каждой модели является довольно существенной. Как и ожидалось, легкая модель градиентного бустинга работает быстрее всего. На втором месте находится экстремальный градиентный бустинг, на третьем – адаптивный бустинг. Построение ансамблевых моделей бэггинга занимает наибольшее количество времени, что сильно идет в ущерб общей эффективности работы алгоритма.

В данной задаче равный приоритет имеют как метрики точности предсказаний, так и метрика затраченного времени. LightGBM хоть и является самой быстрой по исполнению моделью, но допускает большое количество ошибок при классификации. Методы бэггинга: бэггинг и случайный лес показывают хорошую точность в выявлении мошеннических транзакций, но их построение занимает довольно большой временной промежуток, что значительно отразится на всей системе, проводящей транзакции. Результаты работы XGBoost являются наиболее точными, а также этот метод оптимален по затраченному на построение ансамбля времени. То, что XGBoost наилучшая модель для выявления мошеннических транзакций, отражает и реальную картину в финансовых организациях, где в основе таких систем, отслеживающих транзакции, также лежит метод XGBoost.

* 1. Сравнение результатов и метрик оценки результатов

Если рассматривать вместе задачи регрессии и классификации, то нужно учитывать, что в данном случае размеры набора данных сильно различались, поэтому сравнивать время, затраченное на построение, то есть обучение и тестирование моделей, не будет объективно отражать настоящую разницу во временной эффективности ансамблевых моделей. Тем не менее, можно отметить, что в обоих случаях, как и ожидалось, работа оптимизированной модели LightGBM заняла меньше всего времени.

При сравнительном анализе точности предсказаний ансамблевых моделей в обоих случаях упомянутая модель легкого градиентного бустинга при этом в результате дала наихудшие метрики, значит, быстрота исполнения метода оказалась в ущерб точности предсказаний. Другие модели бустинга, адаптивный и экстремальный градиентный, показали различные результаты в зависимости от решаемой задачи.

Кроме того, при решении обеих прикладных задач метод бэггинга показал большую эффективность, так как алгоритм бутстрэпинга при формировании выборок для базовых алгоритмов помог решить проблемы в используемых датасетах: при предсказании суммы потерь компании в случае утечки данных бутстрэпинг помог повысить эффективность модели на небольшом наборе данных, а в задаче выявления мошеннических транзакций данный алгоритм способствовал решению проблемы несбалансированности классов, так как при формировании выборок для «слабых учеников» фродовые транзакции могли попасть в выборку несколько раз.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы был проведен сравнительный анализ ансамблевых методов машинного обучения: бэггинга, бустинга и случайного леса. Каждый метод был представлен одной или более разновидностями моделей со своими особенностями и параметрами.

Методы бэггинга и случайного леса были представлены моделями Bagging и Random Forest. Данный метод подразумевает независимое обучение базовых алгоритмов, набор элементов для обучения которых формируется методом бутстрэп, что обеспечивает большую независимость деревьев.

Другой исследуемый ансамблевый метод деревьев решений – бустинг представляет собой последовательное обучение базовых алгоритмов, учитывая в последующих «слабых учениках» ошибки предыдущих. Используемыми моделями бустинга были адаптивный бустинг AdaBoost и две модели градиентного бустинга XGBoost и LightGBM.

Ансамблевые методы были применены в двух видах прикладных задач из области информационной безопасности: задача регрессии представляла собой предсказание суммы потерь организации при утечке данных, а задачей классификации была избрана задача выявления мошеннических транзакций среди общего числа транзакций, проходящих через систему компании. Таким образом, был произведен более полный сравнительный анализ исследуемых методов, что позволило получить лучшее представление об эффективности работы ансамблевых методов на разных типах задач.

Сравнительный анализ производился по определенным в работе метрикам оценки эффективности, в частности метрики точности предсказания, уникальные для каждого типа задач, а также метрика затраченного на построение и исполнения модели времени.

В результате сравнительного анализа было выявлено, что в задаче предсказания суммы потерь компании из-за утечки данных наилучшие результаты точности предсказаний показали модели бэггинга и адаптивного бустинга. Однако, взяв во внимание то, что модель адаптивного бустинга не была эффективной по затраченному времени, метод бэггинга был выбран наиболее подходящим для решения данной задачи.

При исследовании задачи классификации мошеннических транзакций наибольшая точность предсказаний наблюдалась у модели XGBoost, а также хорошая точность в моделях бэггинга: Bagging и Random Forest. XGBoost в данном случае был выбран наилучшей моделью, так как показал одно из наименьших значений затраченного времени.

Рассмотрев вместе две прикладные задачи, можно сказать, что для обоих наборов данных метод бэггинга показал хорошие результаты как по точности предсказаний, так и по затраченному на построение модели времени. Это можно объяснить тем, что метод формирования выборок для базовых учеников в бэггинге – бутстрэпинг – способствовал борьбе с проблемами в используемых датасетах. Также, для обеих задач было равным то, что модель легкого градиентного бустинга LightGBM, хоть и показала наименьшее затраченное время, являлась не эффективной и не отвечала желаемым метрикам оценки точности предсказания, соответственно, при использовании этой модели есть большой риск ошибки, который может не стоить сохраненного времени.

Можно сказать, что заданная цель работы была достигнута: были проведены сравнительный анализ ансамблевых методов деревьев решений при решении задач регрессии и классификации и исследование метрик эффективности работы построенных моделей. Также в конце работы приведены выводы и результаты данного сравнительного анализа.

В качестве перспективы дальнейшего исследования ансамблевых методов деревьев решений возможно изучение других моделей ансамблевых методов, а также их комбинации. Представляется возможным провести оптимизацию подбора параметров для моделей, которое на данный момент не является задачей работы, но занимает определенное время. Наконец, возможно исследование эффективности ансамблевых методов деревьев решений при решении других прикладных задач на отличающихся наборах данных, в этом случае, метрики и лучшие модели могут отличаться от тех, что были получены в работе.

Ансамблевые методы деревьев решений могут быть сильным и качественным инструментом для решения задач как в области информационной безопасности, так и в других сферах. Важно учитывать специфику задачи и набора данных, чтобы подобрать метод, который будет отвечать требованиям как точности предсказаний значений, так и требованиям эффективности исполнения. При верном подборе метода и параметров ансамблевой модели можно получить модель, эффективно решающую поставленную задачу.

Изображение выглядит как текст

Автоматически созданное описаниеДанная работа выполнена мною самостоятельно.

«25» мая 2022 г. \_\_\_\_\_\_\_\_\_\_\_

СПИСОК ИСТОЧНИКОВ

1. *Opitz D., Maclin R.* Popular ensemble methods: An empirical study // Journal of Artificial Intelligence Research. — 1999
2. *Левитин А. В.* Глава 10. Ограничения мощи алгоритмов: Деревья принятия решения // Алгоритмы. Введение в разработку и анализ — М.: Вильямс, 2006. — С. 409—417. — 576 с.
3. *Breiman, L.* Bagging Predictors // Machine Learning. — 1996. — Т. 24, вып. 2.
4. *Hastie, T., Tibshirani R., Friedman J.* Chapter 15. Random Forests // The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed. — Springer-Verlag, 2009. — 746 с.
5. *Kuncheva L., Whitaker C.* Measures of diversity in classifier ensembles and Their Relationship with the Ensemble Accuracy // Machine Learning. — 2003. — Т. 51, вып. 2.
6. *Rokach L.* Ensemble-based classifiers // Artificial Intelligence Review. — 2010. — Т. 33, вып. 1—2.
7. Чжи-Хуа Чжоу Ансамблевые методы: 2012.
8. Сергей Николенко Деревья принятия решений // machine learning - cs club, 2018.
9. Мошков М.Ю. Деревья решений. Теория и приложения: М.: Букинист, 1994.
10. Де Прадо М. Машинное обучение: алгоритмы для бизнеса: . - СПб.: Питер, 2019. - 432 .
11. Cha Zhang, Yunqian Ma Ensemble Machine Learning: Methods and Applications: 2012
12. Jason Brownlee Ensemble Learning Algorithms With Python: Make Better Predictions with Bagging, Boosting, and Stacking: 2021. 450 с.
13. Джордж Кириакидес Hands‑On Ensemble Learning with Python : PacktPublishing, 2019.
14. Decision Trees in Machine Learning // https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
15. Color Guide to Seaborn Palettes // https://medium.com/@morganjonesartist/color-guide-to-seaborn-palettes-da849406d44f
16. Bagging and Random Forest for Imbalanced Classification // https://machinelearningmastery.com/bagging-and-random-forest-for-imbalanced-classification/
17. Measuring Performance: AUPRC and Average Precision // https://glassboxmedicine.com/2019/03/02/measuring-performance-auprc/
18. A Gentle Introduction to the Fbeta-Measure for Machine Learning // https://machinelearningmastery.com/fbeta-measure-for-machine-learning/
19. XGBoost Documentation // https://xgboost.readthedocs.io/en/latest/index.html
20. LightGBM’s documentation // https://lightgbm.readthedocs.io/en/latest/index.html
21. Scikit-learn Documentation // https://scikit-learn.org/
22. Деревья решений: общие принципы // https://loginom.ru/blog/decision-tree-p1
23. A Guide to AdaBoost // https://blog.paperspace.com/adaboost-optimizer/
24. Decision Trees: Understanding the Basis of Ensemble Methods // https://towardsdatascience.com/decision-trees-understanding-the-basis-of-ensemble-methods-e075d5bfa704

ПРИЛОЖЕНИЕ

1. Файл .ipynb «Аблязова Алия ПМ18-1»