Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве**

**Российской Федерации»**

Департамент анализа данных и машинного обучения

Курсовая работа по дисциплине

«Технологии анализа данных и машинного обучения»

на тему:

**«Леса деревьев решений в задачах регрессии»**

Выполнила:

студентка группы ПИ20-4

Еркебаева А. Д.

Научный руководитель:

профессор, д.э.н. Демин И. С.

**Москва**

**2023**

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_Toc135239459)

[ОСНОВНАЯ ЧАСТЬ 5](#_Toc135239460)

[Глава 1. Описание предметной области и алгоритма реализации 5](#_Toc135239461)

[Описание понятия регрессия 5](#_Toc135239462)

[1.2 Описания алгоритма решения на основе лесов деревьев решений 7](#_Toc135239463)

[1.3 Описание методов оценки лесов деревьев решений 11](#_Toc135239464)

[Глава 2. Разведочный анализ данных 13](#_Toc135239465)

[Глава 3. Предобработка данных 19](#_Toc135239466)

[3.1. Очистка данных 19](#_Toc135239467)

[3.2 Преобразование данных 20](#_Toc135239468)

[3.3 Разделение данных 22](#_Toc135239469)

[Глава 4. Реализация алгоритмов и обучение моделей. 23](#_Toc135239470)

[4.1 RandomForestRegressor без указания гиперпараметров 23](#_Toc135239471)

[4.2 RandomForestRegressor с RandomizedSearchCV 24](#_Toc135239472)

[4.3 DecisionTreeRegressor без гиперпараметров 26](#_Toc135239473)

[4.4 DecisionTreeRegressor с RandomizedSearchCV 27](#_Toc135239474)

[4.5 Глубокая нейронная сеть прямого распространения 28](#_Toc135239475)

[4.6 Сравнение моделей 30](#_Toc135239476)

[ЗАКЛЮЧЕНИЕ 32](#_Toc135239477)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 33](#_Toc135239478)

[ПРИЛОЖЕНИЕ 34](#_Toc135239479)

# ВВЕДЕНИЕ

Леса деревьев решений являются одним из наиболее популярных методов машинного обучения, используемых для решения задач классификации и регрессии. В данной курсовой работе будет рассмотрена тема лесов деревьев решений в задачах регрессии. Будут рассмотрены основные принципы работы данного метода, а также его преимущества и недостатки. Кроме того, будет проведен анализ эффективности использования лесов деревьев решений в задачах регрессии на примере реальных данных и сравнение с применением глубокого машинного обучения с помощью библиотеки PyTorch.

Гостиничный бизнес и туризм являются отраслями с высокой степенью конкуренции. Одной из задач, которую нужно решать в этих отраслях, является предсказание цен на гостиничные номера, авиабилеты и другие услуги. Лес деревьев решений может помочь решить эту задачу.

Актуальность данной темы обусловлена необходимостью прогнозирования различных показателей в гостиничном бизнесе и туризме, таких как количество гостей, популярность отелей, сезонность и т.д. Использование лесов деревьев решений позволяет получать более точные прогнозы благодаря способности учитывать множество факторов и их взаимодействие. Это может помочь компаниям в принятии более обоснованных решений, оптимизации бизнес-процессов и увеличении прибыли. Кроме того, данная тема актуальна в свете тенденций развития технологий и применения анализа данных в различных отраслях экономики.

Цель исследования состоит в том, чтобы изучить применения лесов деревьев решений в задачах регрессии, то есть предсказать численные значения целевой переменной.

Исследование будет включать в себя рассмотрение различных методов построения и обучения лесов деревьев решений, таких как RandomForestRegressor, DecisionTreeRegressor, а также модель нейронной сети глубокого обучения.

В гостиничном бизнесе и туризме, лес деревьев решений могут использоваться для решения различных задач, таких как:

* Предсказание цен на гостиничные номера: на основе различных факторов, таких как дата пребывания, услуги, расположение и т. д., может быть построен массив деревьев, который помогает предсказать цену на гостиничный номер.
* Рекомендации по маршруту: на основе предпочтений клиента, доступных ресурсов и других факторов, лес деревьев решений может помочь подобрать маршрут для клиента.
* Определение потенциальных клиентов: лес деревьев решений может использоваться для определения потенциальных клиентов, которые могут быть заинтересованы в услугах гостиничного бизнеса и туризма.

Основными преимуществами лесов деревьев решений являются их способность работать с большим количеством признаков и обрабатывать данные с различными типами переменных. Кроме того, этот метод не требует предварительной обработки данных, такой как масштабирование или нормализация, что упрощает процесс моделирования. Еще одним преимуществом лесов деревьев решений является их способность обрабатывать данные с пропущенными значениями.

# ОСНОВНАЯ ЧАСТЬ

## Глава 1. Описание предметной области и алгоритма реализации

### Описание понятия регрессия

Регрессия — это метод статистического анализа, который используется для изучения зависимостей между двумя или более переменными. Он позволяет определить, как одна переменная воздействует на другую переменную. В регрессионном анализе оценивается степень взаимосвязи между зависимой переменной (такой как доход, продажи или цена акций) и независимой переменной (такой как расходы на рекламу, процентные ставки или количество произведенных товаров).

Регрессия может быть линейной, когда связь между переменными представляется прямой линией, или нелинейной - когда связь более сложная и не может быть описана прямой линией. Регрессионный анализ используется в различных областях, включая экономику, финансы, маркетинг, социологию, биологию и многие другие.

Регрессионный анализ – это метод статистического анализа, который используется для измерения отношения между зависимой переменной и одной или несколькими независимыми переменными.

Регрессия обучается путем поиска оптимальной функции, которая лучше всего описывает зависимую переменную в зависимости от независимых переменных. Она может быть обучена на основе известного набора данных, где значения независимых и зависимых переменных уже известны, или на основе набора данных, где значения только независимых переменных известны, а значения зависимой переменной требуются для прогнозирования.

Существует несколько методов регрессионного анализа, которые могут быть использованы для прогнозирования значений зависимой переменной.

1. Простая линейная регрессия (Simple Linear Regression) - используется для прогнозирования значения зависимой переменной на основе одной независимой переменной.

2. Множественная линейная регрессия (Multiple Linear Regression) - используется для прогнозирования значения зависимой переменной на основе нескольких независимых переменных.

3. Логистическая регрессия (Logistic Regression) - используется для прогнозирования вероятности бинарного исхода зависимой переменной.

4. Гребневая регрессия (Ridge Regression) - используется для регуляризации линейной регрессии и предотвращения переобучения.

5. LASSO регрессия (Lasso Regression) - также используется для регуляризации линейной регрессии, но она также помогает идентифицировать наиболее важные переменные для моделирования.

6. Elastic Net регрессия (Elastic Net Regression) — это комбинация гребневой регрессии и LASSO регрессии.

7. Робастная линейная регрессия (Robust Regression) - используется для минимизации влияния выбросов на модель.

8. Леса деревьев решений регрессор (Random Forest Regressor) Основная идея леса деревьев решений состоит в создании множества деревьев решений, где каждое дерево обучается независимо от других на подмножестве данных и случайных подмножествах признаков. При этом каждое дерево решает задачу регрессии для предсказания целевой переменной.

Выбор метода регрессионного анализа зависит от типа данных и прогнозируемой переменной.

### 1.2 Описания алгоритма решения на основе лесов деревьев решений

Дерево решений строится путем разбиения данных на все более мелкие подгруппы, которые имеют одинаковые характеристики. Для этого используется алгоритм, который выбирает наилучшее разделение на каждом узле дерева, используя выбранный критерий разделения. На каждом узле дерева анализируется каждый признак и выбирается тот, который наилучшим образом разделяет данные на две группы. Этот процесс продолжается до тех пор, пока не будет достигнут критерий остановки, такой как достижение максимальной глубины дерева или минимального количества объектов в листе. Когда дерево решений построено, оно может быть использовано для классификации или прогнозирования новых данных.

Для реализации лесов деревьев решений в задачах регрессии необходимо выполнить следующие шаги:

1. Подготовить данные. Необходимо выбрать признаки, которые будут использоваться для построения модели, а также выполнить обработку данных, такую как заполнение пропущенных значений.

2. Разбить данные на обучающую и тестовую выборки. Обучающая выборка будет использоваться для построения модели, а тестовая выборка – для проверки ее качества.

3. Построить лес деревьев решений. Для этого необходимо задать параметры модели, такие как количество деревьев, глубина каждого дерева и т. д.

4. Оценить качество модели на тестовой выборке. Для этого можно использовать различные метрики, такие как средняя абсолютная ошибка или коэффициент детерминации.

5. При необходимости настроить параметры модели для улучшения ее качества.

В лесу деревьев решений каждое дерево строится на основе случайной выборки данных и случайного набора признаков. Это позволяет уменьшить вероятность переобучения модели и повысить ее обобщающую способность.

При прогнозировании значения целевой переменной лес деревьев решений проводит агрегацию прогнозов всех деревьев, например, путем усреднения или голосования. Это позволяет получить более точный прогноз, чем при использовании отдельного дерева решений.

Изображение выглядит как диаграмма, белый, линия, зарисовка

Автоматически созданное описание

Рисунок 1 – Визуализация дерева решений

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 2 – Визуализация дерева решений текстом

Алгоритмы построения деревьев решений могут быть разделены на две категории:

1. Рекурсивное разбиение данных — это метод, который используется для разбиения данных на более мелкие группы, начиная с корня дерева и двигаясь вниз по его ветвям. Этот метод использует некоторую метрику для определения того, какие параметры следует разделить.

2. Обучение на основе правил — это метод, который обучает модель на основе правил на основе данных. Этот метод может использоваться для построения деревьев, которые учитывают разные факторы при создании прогнозов.

Одной из главных проблем с деревьями решений является их склонность к переобучению. Другим словами, они могут производить хорошие результаты на тренировочных данных, но плохо работать на новых данных. Для того, чтобы избежать этой проблемы, могут использоваться методы регуляризации, такие как ограничение глубины дерева или ограничение числа листьев в дереве.

Преимуществами алгоритмов деревьев решений являются их простота и интуитивная понятность. Они также могут использоваться как в задачах классификации, так и в задачах регрессии, а также в задачах выживания. Они могут легко обрабатывать нечисловые данные, такие как категории и текст, и могут использоваться для создания графических моделей, которые легко интерпретировать.

Существуют различные алгоритмы деревьев решений:

1. Алгоритм ID3 - алгоритм построения дерева решений, который выбирает наилучший признак для разбиения на каждом шагу, используя энтропию.

2. Алгоритм C4.5 - улучшенная версия алгоритма ID3, которая учитывает неполноту данных и имеет возможность работать с категориальными и числовыми признаками.

3. Алгоритм CART - алгоритм, который использует индекс Джини для выбора наилучшего признака на каждом шаге. Он может обрабатывать как задачи классификации, так и регрессии.

4. Random Forest - алгоритм, который использует множество решающих деревьев для улучшения точности классификации. Каждое дерево построено на подмножестве обучающей выборки и случайном выборе признаков.

5. Gradient Boosting Tree - алгоритм, который строит последовательность деревьев, каждое из которых исправляет ошибки предыдущего. Он использует градиентный спуск для нахождения оптимальных весов деревьев.

6. XGBoost - улучшенная версия Gradient Boosting Tree, которая имеет множество оптимизаций и регуляризаций для улучшения точности и скорости работы. Он используется во многих соревнованиях по машинному обучению.

Основные параметры дерева решений включают:

1. Критерий разбиения: критерий, который используется для выбора лучшего разбиения на каждом узле дерева.

2. Максимальная глубина дерева: максимальное количество разбиений (уровней) в дереве.

3. Минимальное количество объектов в листе: минимальное количество объектов, которые должны находиться в каждом листовом узле дерева.

4. Максимальное количество листьев: максимальное количество листьев в дереве.

5. Минимальное уменьшение неопределенности: минимальное значение прироста информации, которое должно произойти при разбиении данных на каждом узле дерева.

6. Максимальное количество признаков: максимальное количество признаков, которые могут быть использованы при каждом разбиении.

Эти параметры влияют на структуру дерева, его способность к обобщению данных и склонность к переобучению. Оптимальные значения параметров должны быть подобраны для каждого конкретного случая.

### 1.3 Описание методов оценки лесов деревьев решений

Оценки для лесов деревьев решений в основном основываются на точности классификации или регрессии. Для регрессии возможно использование метрик, таких как:

1. Mean Absolute Error (MAE) - средняя абсолютная ошибка, которая измеряет среднее абсолютное отклонение прогнозируемых значений от фактических значений.

2. Mean Squared Error (MSE) - средняя квадратичная ошибка, которая измеряет среднее квадратичное отклонение прогнозируемых значений от фактических значений.

3. Root Mean Squared Error (RMSE) - корень из средней квадратичной ошибки, которая измеряет среднее квадратичное отклонение прогнозируемых значений от фактических значений.

4. R-squared (R2) - коэффициент детерминации, который измеряет долю объясненной дисперсии в общей дисперсии целевой переменной. Чем ближе значение R2 к 1, тем лучше модель.

5. Mean Absolute Percentage Error (MAPE) - средняя абсолютная процентная ошибка, которая измеряет среднее абсолютное отклонение прогнозируемых значений от фактических значений в процентах.

Иногда также может быть использована кросс-валидация для оценки качества модели на нескольких независимых наборах данных.

Кросс-валидация — это метод, который позволяет оценить качество работы лесов деревьев решений путем разбиения выборки на несколько частей и последовательного использования каждой из них в качестве тестовой выборки. Кросс-валидация позволяет уменьшить искажение оценки качества, вызванное случайностью разбиения выборки.

## Глава 2. Разведочный анализ данных

Разведочный анализ данных — это процесс изучения данных с целью выявления закономерностей, тенденций, отклонений и других важных характеристик в них. EDA является первым этапом в анализе данных и помогает исследователю понять, что содержится в наборе данных, какие параметры наиболее значимы и как эти параметры связаны между собой. Он позволяет проводить предварительную оценку данных и делать первые выводы о том, что можно будет извлечь из этих данных. Основные цели разведочного анализа данных:  
- Исследование данных и выявление закономерностей;  
- Оценка качества данных (выявление выбросов, пропущенных данных и т. п.);  
- Поиск связей между различными признаками;  
- Выбор наиболее значимых признаков для построения модели;  
- Описание данных.

В данной курсовой работе используется датасет Hotel Booking Demand, который содержит информацию о бронировании городского и курортного отелей, целевой переменной в данном случае будет являться средняя дневная ставка.

В датасете более 110 тысяч строк данных и 32 столбца. Присутствуют пустые значения, замена которых будет описана в следующей главе о предобработке данных.

Для дальнейшего анализа по графикам в отдельную копию датасета были выделены только данные, бронирование которых не отменено, то есть столбец is\_canceled = 0.

На рисунке 3 можно увидеть гистограмму распределения количества гостей по странам, из которых они прибыли. По нему можно заметить, что больше всего гостей прибыло из Португалии, Испании и Италии.

На рисунке 4 отображена более подробная информация по гистограмме «Страна – количество гостей» с группировкой по типу отеля, чтобы точнее понять какие типы отелей выбирают гости из разных стран.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 – Гистограмма «Страна – количество гостей» Изображение выглядит как текст, снимок экрана, График, линия

Автоматически созданное описание

Рисунок 4 – График

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Рисунок 5 – График «Месяц – количество гостей»

На рисунке 5 изображен график зависимости количества гостей отелей от месяца. По нему можно заметить, что отели пользуются наибольшим спросом в августе, а наименьшим – в январе.

Изображение выглядит как текст, снимок экрана, диаграмма, График

Автоматически созданное описание

Рисунок 6 – Ящик с усами «Тип комнаты – ставка» в зависимости от типа отеля

Для того, чтобы понять, насколько сильный разброс в данных по ставке в зависимости от типа комнаты и типа отеля был выбран график ящик с усами, который изображен на рисунке 6. На нем можно заметить, что сильный выбросы присутствуют и что их необходимо нормализовать для дальнейшего обучения.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 7 – Гистограмма «Месяц – количество гостей» в зависимости от типа отеля

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

Рисунок 8 – График «Месяц – количество гостей» в зависимости от типа отеля

На рисунках 7 и 8 изображены график и гистограмма зависимости количества гостей от месяца посещения в разрезе типа отеля. Судя по обоим графикам, во все месяца городские отели посещают больше, чем курортные, однако, например, в январе и декабре эта разница наименьшая.

Изображение выглядит как снимок экрана, диаграмма, текст, Графика

Автоматически созданное описание

Рисунок 9 – Круговая диаграмма посещений по годам

На круговой диаграмме (рис. 9) мы видим процентное соотношения количества гостей по годам. И судя по этой диаграмме, можно понять, что посещаемость отелей с каждым годом значительно увеличивалась почти в 2 раза.

Изображение выглядит как диаграмма, снимок экрана, круг, Графика

Автоматически созданное описание

Рисунок 10 – Круговая диаграмма выбора типа питания

На круговой диаграмме на рисунке 10 показано процентное соотношения выбора типа питания. Видно, что почти 75% гостей выбирают тип питания «BB», то есть включают в свое проживание только завтрак, что более распространено как раз в городских отелях, а по графику с рисунка 8, мы уже узнали, что большинство гостей выбирают именно городской отель. Наименьшее же предпочтение (менее 1%) гости отдают типу «FB», что соответствует трехразовому питанию – завтрак, обед, ужин.

Изображение выглядит как диаграмма, снимок экрана, Графика, круг

Автоматически созданное описание

Рисунок 11 – Круговая диаграмма посещений

По круговой диаграмме с рисунка 11, можно заметить, что достаточно большой процент бронирований (более 38%) в итоге отменяется. Для отеля большой процент отмены брони означает потерю потенциальных доходов и возможность увеличения числа свободных номеров. Кроме того, это может указывать на проблемы с сервисом, комфортом или ценовой политикой отеля, которые могут отталкивать гостей от бронирования номеров.

## Глава 3. Предобработка данных

Предобработка данных — это процесс подготовки данных к использованию в анализе и моделировании. Она включает в себя различные действия над данными для очистки, преобразования, исправления ошибок или устранения пропущенных значений.

В самом начале выводим heatmap (рис. 12), чтобы определить зависимости признаков друг от друга.

Изображение выглядит как снимок экрана, шаблон, красный, Красочность

Автоматически созданное описание

Рисунок 12 – heatmap

### 3.1. Очистка данных

Как было сказано в предыдущей главе, в качестве целевой переменной был выбран столбец adr, который хранит среднюю дневную ставку. Значит корреляцию остальных признаков мы будем считать именно по этому признаку.

Леса деревьев решений могут обрабатывать коррелированные переменные, так как они используют случайные выборки признаков для каждого дерева в лесу. Это означает, что каждое дерево оценивает признаки независимо от других деревьев, что уменьшает корреляцию между признаками.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 13 – Расчет корреляции

Анализируя полученные данные, можно исключить из датасета последние 8 столбцов списка, так как они почти не влияют на наши данные и не будут полезны нам в дальнейшем анализе.

### 3.2 Преобразование данных

Далее создаются два отдельных датасета для категориальных и числовых столбцов. Столбец reservation\_status\_date разбивается на 3 разных столбца: день, месяц, год, для более удобной обработки. Далее все категориальные столбцы с помощью LabelEncoder трансформируются в числовые (рис. 14), так как в регрессионных моделях используются именно числовые признаки. В датафрейме с числовыми признаками проверяется дисперсия каждого признака в зависимости от целевого (рис. 15), и к столбцам, где слишком большое значение дисперсии, применяется min-max нормализация (рис. 16).

Изображение выглядит как текст, снимок экрана, число, Параллельный

Автоматически созданное описание

Рисунок 14 – Трансформированные категориальные данные

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 15 – Дисперсия

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 16 – Итоговые дисперсии

### 3.3 Разделение данных

Выборка делится с помощью train\_test\_split на обучающую и тестовую в отношении 2:8 для оценки качества работы алгоритма машинного обучения. Обучающая выборка используется для обучения модели, а тестовая выборка для тестирования ее точности на данных, которые алгоритм не видел в процессе обучения. Это позволяет оценить, насколько хорошо модель обобщает данные и применяется к новым данным вне обучающего набора. Если выборка не делится, то можно получить переобучение модели, когда она работает хорошо на данных обучения, но плохо на новых данных.

## Глава 4. Реализация алгоритмов и обучение моделей.

В начале определяем функции show\_metrics для отображения метрик модели, в нашем случае выбраны MSE и R^2, и find\_hyper для поиска лучших гиперпараметров модели с помощью RandomizedSearchCV.

### 4.1 RandomForestRegressor без указания гиперпараметров

RandomForestRegressor — это алгоритм машинного обучения, который используется для регрессионного анализа. Он построен на базе двух основных концепций: случайного леса и регрессии.

Случайный лес представляет собой ансамбль решающих деревьев. Каждое дерево в лесу строится на основе подмножества обучающей выборки и произвольно выбранных факторов. Затем, для прогнозирования этого дерева, производится суммирование результатов всех деревьев в лесу. Это позволяет улучшить точность прогнозирования и снять многие ограничения, связанные с использованием одного дерева для прогнозирования.

Для обучения модели Random Forest Regressor данные разбиваются на меньшие части, и каждая часть обучается на отдельном дереве. Затем, для каждой записи из тестового набора данных, модель использует все деревья для определения прогнозируемого значения.

Преимуществами Random Forest Regressor являются высокая точность прогнозирования и способность работать с большими объемами данных. Однако он также может страдать от ошибок при работе с несбалансированными данными, недостатком интерпретации принятых решений и медленной скорости работы.

R2\_score (коэффициент детерминации) показывает, насколько хорошо линейная регрессия приближает данные. Значение r2\_score варьируется от 0 до 1, где 1 означает идеальное совпадение данных с моделью, а 0 означает, что модель не приближает данные лучше, чем простое среднее значение.

MSE (среднеквадратическая ошибка) — это среднее значение квадрата ошибки между прогнозируемыми значениями модели и реальными значениями. Значение MSE показывает, насколько хорошо модель приближает данные: чем меньше значение, тем лучше модель. Ошибка в квадрате применяется, чтобы выдать большой штраф за значительные отклонения.

При проверке на тестовой выборке R^2 модели составляет 0,9, что означает что модель может предсказать значение с 90% точностью, а ошибка 229,5 (рис. 17).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 17 – Метрики для RandomForestRegressor без указания гиперпараметров

### 4.2 RandomForestRegressor с RandomizedSearchCV

RandomizedSearchCV (Randomized Search Cross-Validation) это метод поиска оптимального набора параметров для модели машинного обучения. Он является альтернативой GridSearchCV, который перебирает все возможные комбинации параметров, что может быть очень долго при большом количестве параметров. В отличие от GridSearchCV, RandomizedSearchCV выбирает случайным образом наборы параметров для оценки производительности модели. Это позволяет уменьшить время перебора параметров и способствует более быстрому поиску оптимальных параметров.

В данной модели в качестве гиперпараметров, которые мы собираемся проверять, берем:

* N\_estimators – количество деревьев в лесу
* Max\_depth – максимальная глубина дерева
* Min\_samples\_split – минимальное количество выборок, для разделения узла
* Min\_samples\_leaf – минимальное количество выборок, которые будут находиться в конечно узле
* Max\_features – количество учитываемых функций

В данном случае у модели с подбором гиперпараметров для RandomForestRegressor (рис. 18) с помощью RandomizedSearchCV получились метрики чуть хуже, но все еще модель обучилась успешно, так как R^2 на тестовом множестве 0,89 (рис. 19).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 18 – RandomizedSearchCV для RandomForestRegressor

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 19 – Метрики RandomForestRegressor c RandomizedSearchCV

### 4.3 DecisionTreeRegressor без гиперпараметров

DecisionTreeRegressor — это модель регрессии, основанная на деревьях решений. Она предсказывает значения целевой переменной, разбивая данные на более мелкие подмножества на основе значений признаков.

Дерево решений имеет корень, ветви и листья. Каждый узел решения представляет собой вопрос о значении одного признака, а каждая ветвь проходит через возможные значения ответа на этот вопрос. Листовые узлы решения представляют собой прогнозы.

DecisionTreeRegressor строит дерево решений, используя алгоритм ID3 (Iterative Dichotomiser 3). Алгоритм начинается с корневого узла и разбивает выборку, выбирая признак с наилучшей информационной выгодой. Итерация повторяется для каждого дочернего узла до тех пор, пока не будет достигнут критерий останова, например, когда количество объектов в узле меньше определенного порога или значения метрики ошибки в узле слишком малы.

Когда модель показывает очень высокое значение R-квадрат на тренировочном наборе данных (0,99), но низкое значение на тестовом наборе данных (0,8), это может быть свидетельством переобучения модели на тренировочных данных (рис. 20).

Переобучение — это процесс, при котором модель становится слишком комплексной и начинает "запоминать" тренировочные данные вместо того, чтобы обобщать их и применять для предсказания на новых данных. В таком случае, когда модель сталкивается с новыми данными, которые она не "запоминала" на тренировке, ее точность падает.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 20 – DecisionTreeRegressor без гиперпараметров

### 4.4 DecisionTreeRegressor с RandomizedSearchCV

В данной модели мы устанавливаем, что можем выбирать значения для таких гиперпараметров, как:

* Criterion – функция для оценки качества разделения
* Max\_depth – максимальная глубина дерева
* Min\_samples\_split – минимальное количество выборок, которое нужно в конечном узле
* Max\_leaf\_nodes используется для ограничения количества листовых узлов в дереве решений при обучении модели
* Max\_features – количество функций, которые следует учитывать при поиске наилучшего разделения:

На основании полученных метрик можно сделать вывод, что модель достаточно хорошо обучилась, метрики достаточно высокие, так как r^2 почти равен 0,8, что демонстрирует хорошее качество модели, но модель все же может иметь небольшие неточности.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 21 – DecisionTreeRegressor с RandomizedSearchCV

Изображение выглядит как текст, снимок экрана, Шрифт, чек

Автоматически созданное описание

Рисунок 22 – Метрики DecisionTreeRegressor с RandomizedSearchCV

### 4.5 Глубокая нейронная сеть прямого распространения

Нейронная сеть состоит из пяти полносвязных слоев, каждый из которых содержит набор нейронов, которые принимают входные значения от предыдущего слоя, умножают их на соответствующие им веса, производят линейную комбинацию и отправляют дальше после применения функции активации ReLU и написана с помощью применения библиотеки pytorch.

В качестве критерия, который измеряет ошибку, был взят MSELoss, а оптимизатором выбран Adam. Количество эпох для обучения – 60. По графику (рис. 24) мы видим, что модель достаточно хорошо обучалась, а по полученным метрикам (рис. 25), что модель на тестовом множестве даже показала себя лучше, чем на тренировочном, что показывает, что она научилась обобщать и предсказывать данные достаточно хорошо.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 23 – Обучение модели

=Изображение выглядит как текст, снимок экрана, График, линия

Автоматически созданное описание

Рисунок 24 – График функции потерь

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 25 – Метрики модели

### 4.6 Сравнение моделей

В ходе исследования было применено 5 различных решений для нахождения наиболее подходящего и наиболее эффективного. В таблице (рис. 26) приведены все метрики для использованных алгоритмов и модели для того, чтобы наглядно увидеть лучшую модель. По r^2 на тренировочном множестве явно выделяется RandomForestRegressor с дефолтными гиперпараметрами, однако на тестовом это все-таки не самый лучший алгоритм. Стоит отметить, что у DecisionTreeRegressor метрики на тренировочном множестве лучше, однако они сильно разнятся с метриками на тестовом, что означает, что модель обучалась больше для своих данных и обобщает на среднем уровне. У RandomForestRegressor с гиперпараметрами, подобранными RandomizedSearchCV лучший показатель на тестовом множестве, что означает, что он лучше остальных моделей может предсказать данные.

Как мы видим, деревья решений по качеству можно сравнить с нейронными сетями глубокого обучения, а иногда они даже их превосходят.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 25 – Таблица всех параметров всех моделей

# ЗАКЛЮЧЕНИЕ

Таким образом, леса деревьев решений – это метод машинного обучения, который может использоваться для эффективного решения задач регрессии. Они являются мощным инструментом для решения задач регрессии и позволяют улучшить качество прогнозов и обеспечить более точные результаты, чем традиционные методы регрессии, на примере было замечено также, что в некоторых случаях они даже превышают по метрикам нейронные сети глубокого обучения.

Когда необходимо сделать прогноз на новых данных, каждое дерево решений в лесу делает свой прогноз, а затем они все объединяются в один окончательный. Обычно используется среднее значение всех прогнозов деревьев. Это позволяет избежать недостатков отдельных деревьев. Одним из преимуществ по сравнению с одиночным деревом решений является возможность вычисления значимости признаков, которая может быть использована для отбора наиболее важных признаков.

Однако, следует помнить, что выбор оптимального числа деревьев и их параметров может быть утомительным и продолжительным процессом. Тем не менее, преимущества лесов деревьев решений в задачах регрессии явно перевешивают недостатки, что делает их одним из наиболее эффективных инструментов машинного обучения.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. 1.10 Деревья решений // scikit-learn URL: [1.10. Деревья решений - scikit-learn](https://scikit-learn.ru/1-10-decision-trees/) (дата обращения: 10.05.2023).
2. Random Forest Regression in Python // GeeksforGeeks URL: https/www.geeksforgeeks.org/random-forest-regression-in-python/ (дата обращения: 10.05.2023).
3. Чистяков Сергей Павлович Случайные леса: обзор // Труды КарНЦ РАН. 2013. №1. URL: https://cyberleninka.ru/article/n/sluchaynye-lesa-obzor (дата обращения: 10.05.2023).
4. Классификационный и регрессионный анализ с деревьями решений // MachineLearningMastery.ru URL: [Классификационный и регрессионный анализ с деревьями решений (machinelearningmastery.ru)](https://machinelearningmastery.ru/https-medium-com-lorrli-classification-and-regression-analysis-with-decision-trees-c43cdbc58054/) (дата обращения: 10.05.2023)
5. Прогнозное моделирование в IBM SPSS Statistics, R и Python: метод деревьев решений и случайный лес. – М.: ДМК Пресс, 2018. – 642 с.: ил.
6. Д. Шервуд. Видеть лес за деревьями. Системный подход для совершенствования бизнес-модели. – М.: Альпина Паблишер, 2012. – 344 с.

# ПРИЛОЖЕНИЕ

Ссылка на исходный код: [AlexaErkb/CourseWork\_ML (github.com)](https://github.com/AlexaErkb/CourseWork_ML/tree/main)