Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РтФ

Школа профессионального и академического образования

ДОПУСТИТЬ К ЗАЩИТЕ ЕРЕД ГЭК

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (Ф.И.О.)

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Приложение для прогноза стоимости на такси

|  |  |
| --- | --- |
| Научный руководитель | подпись |
|  |  |
|  |  |
| Студент группы РИ-400018, Алнажм Язан | подпись |

Екатеринбург

2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РтФ

Кафедра школа профессионального и академического образования

Направление подготовки 09.04.02 Информационные системы и технологии

Образовательная программа 09.04.02/33.01 – Прикладной анализ данных

УТВЕРЖДАЮ

РОП\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**ЗАДАНИЕ**

на выполнение выпускной квалификационной работы

**студента** Алнажм Язан **группы** *РИ-400018*

(фамилия, имя, отчество)

**1. Тема выпускной квалификационной работы** *Приложение для прогноза стоимость на такси.*

Утверждена распоряжением по институту от «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. № \_\_\_

**2. Руководитель** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О., должность, ученое звание, ученая степень)

**3. Исходные данные к работе** *данные из Яндекс такси*

**4. Перечень демонстрационных материалов** *презентация*

**5. Календарный план**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование этапов выполнения работы | Срок выполнения этапов работы | Отметка о выполнении |
| 1. | *1 раздел (глава)* | до 30.04.2024 г. |  |
| 2. | *2 раздел (глава)* | до 15.05.2024 г. |  |
| 3. | *ВКР в целом* | до 27.05.2024 г. |  |

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 (подпись) Ф.И.О.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ дата (подпись)

**6. Выпускная квалификационная работа закончена** « » мая 2024 г. считаю возможным допустить Алнажм Язан к защите его магистерской диссертации в государственной экзаменационной комиссии.

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) Ф.И.О.

**7. Допустить** Алнажм Язан к защите магистерской диссертации в Государственной экзаменационной комиссии (протокол заседания кафедры №\_\_\_ от «\_\_\_» \_\_\_\_\_ 20\_\_г.).

РОП 09.04.04\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* (подпись) Ф.И.О

**РЕФЕРАТ**

Выпускная квалификационная работа бакалавра 60 с., 2 рис., 25 источн.

СБОР ТРЕБОВАНИЙ, ПРОЕКТИРОВАНИЕ, ПРОТОТИПИРОВАНИЕ, РАЗРАБОТКА ПРИЛОЖЕНИЯ

Объект исследования – процесс прогнозирования стоимости поездок на такси.

Цель работы – разработка приложения для прогноза стоимости на такси с использованием данных реального времени, таких как погода и дорожная ситуация.

Область применения полученных результатов – транспортные услуги и управление стоимостью поездок на такси.

Методы исследования: методы генерации идей, методы активации творческого мышления, метод парных сравнений, экспертных оценок, ранжирование.

Результаты работы: разработан прототип приложения, включающий серверную часть, клиентскую часть. Приложение интегрировано с API для сбора данных о погоде и дорожной ситуации, и использует алгоритмы машинного обучения для прогнозирования стоимости поездок на такси.

Выпускная квалификационная работа выполнена в текстовом редакторе Microsoft Word и представлена в электронном формате.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ………………………………………………………………………………………6](#_Toc169528000)

[1 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ…………………………………………………………..9](#_Toc169528001)

[1.1 Функционирование такси в городской транспортной системе…………….9](#_Toc169528002)

[1.2 Влияние цены на запрос такси……………………………………………...10](#_Toc169528003)

[1.3 Факторы, влияющие на цену запроса такси………………………………..11](#_Toc169528004)

[1.3.1 общие факторы…………….................................................................................11](#_Toc169528005)

[1.3.2 Погодные условия………………………………………………………………12](#_Toc169528006)

[1.3.3 Время суток………………………………………………………………………….12](#_Toc169528007)

[1.3.4 Специальные события……………………………………………………………………………………..13](#_Toc169528008)

[1.4 Способы определения цены такси………………………………………….14](#_Toc169528009)

[2 ТЕОРЕТИЧЕСКАЯ ОСНОВА И МЕТОДЫ………………………………………………..15](#_Toc169528010)

[2.1 Способы сбора данных………………………………………………………15](#_Toc169528011)

[2.1.1 Основы теории сбора данных…………………………………………………………………………..15](#_Toc169528012)

[2.1.2 Методы сбора данных 16](#_Toc169528013)

[2.1.3 Преимущества и недостатки различных методов 16](#_Toc169528014)

[2.2 Способы обработки данных 19](#_Toc169528015)

[2.2.1 Очистка данных 19](#_Toc169528016)

[2.2.2 Преобразование данных 21](#_Toc169528017)

[2.2.3 Интеграция данных 23](#_Toc169528018)

[2.2.4 Нормализация 25](#_Toc169528019)

[2.3 Способы защиты модели от переобучения 28](#_Toc169528020)

[2.3.1 Кросс-валидация 28](#_Toc169528021)

[2.3.2 Регуляризация 29](#_Toc169528022)

[2.3.3 Обрезка 30](#_Toc169528023)

[2.4 Алгоритм регрессии 32](#_Toc169528024)

[2.4.1 Линейная регрессия 32](#_Toc169528025)

[2.4.2 Метод наименьших квадратов (МНК) регрессия 33](#_Toc169528026)

[2.4.3 Бэггинг (Bootstrap Aggregating) 34](#_Toc169528027)

[2.4.4 Алгоритм CatBoost 35](#_Toc169528028)

[2.5 оценка модели 37](#_Toc169528029)

[2.5.1 Метрики точности (Accuracy Metrics) 37](#_Toc169528030)

[2.5.2 Метрики ошибок (Error Metrics) 38](#_Toc169528031)

[2.5.3 Коэффициент детерминации (R-squared) 40](#_Toc169528032)

[3 ПРАКТИЧЕСКАЯ ЧАСТЬ 42](#_Toc169528033)

[3.1 Сбор данных 42](#_Toc169528034)

[3.1.1 Получение цены на такси 42](#_Toc169528035)

[3.1.2 Получение информации о погоде 42](#_Toc169528036)

[3.1.3 Получение информации о дорожной ситуации 43](#_Toc169528037)

[3.1.4 Формирование данных 44](#_Toc169528038)

[3.2 Обработка данных 45](#_Toc169528039)

[3.3.1 Очистка данных 45](#_Toc169528040)

[3.2.2 Преобразование данных 46](#_Toc169528041)

[3.3 Выбор алгоритма 47](#_Toc169528042)

[3.3.1 Реализация алгоритмов 47](#_Toc169528043)

[3.3.2 Оценка и выбор алгоритма 48](#_Toc169528044)

[3.4 Создание блока fastAPI 49](#_Toc169528045)

[3.5 Создание интерфейса React Native 51](#_Toc169528046)

[ЗАКЛЮЧЕНИЕ 54](#_Toc169528047)

[СПИСОК ИСТОЧНИКОВ 57](#_Toc169528048)

# ВВЕДЕНИЕ

В рамках данного исследовательского проекта наше внимание сосредоточено на разработке инновационного приложения, целью которого является предоставление пользователям возможности принимать обоснованные решения при заказе такси. Внутренняя непредсказуемость тарифов такси часто ведет к неопределенности среди пользователей, особенно при выборе оптимального времени для размещения заказа. Наше приложение решает эту проблему, предсказывая потенциальные колебания цен на такси в ближайшие 30 минут, что позволяет пользователям принимать своевременные и экономичные решения.

Значимость нашей работы становится очевидной в контексте улучшения опыта пользователя и оптимизации экономической эффективности. Учитывая частоту заказов такси, предсказательные возможности нашего приложения становятся бесценными, помогая пользователям принимать оптимальные решения. Используя данные в реальном времени и прогностические алгоритмы, наше приложение нацелено на предоставление пользователям ценных исследований будущих тенденций цен на такси.

Кроме того, наше исследование уделяет внимание потенциальным финансовым последствиям для пользователей. Анализируя исторические данные и прогнозируя изменения цен, наше приложение имеет потенциал помочь пользователям сэкономить значительную сумму на такси. Это не только повышает удобство для пассажиров, но также гарантирует, что они смогут получить наилучшие тарифы, размещая заказы в подходящий момент.

В итоге наше исследование разработки приложения для прогнозирования цен на такси крайне актуально, так как оно непосредственно решает распространенную проблему неопределенности в тарифах.

Снабжая пользователей предсказательными данными, наша цель — революционизировать опыт заказа такси, обеспечивая возможность принимать обоснованные решения, приводящие к экономии затрат и повышению комфорта, учитывая частоту заказов такси. Екатеринбуржцы посчитали, что за год цены поднялись в полтора-два раза.

— Я чувствую раздражение из-за того, что подорожало такси. Качество услуг не сильно поменялось, но теперь мы переплачиваем, и я не понимаю, за что. На сумму в 700 рублей можно пообедать или купить ужин на семью из двух-трех человек, а те же деньги уходят на такси за 10 минут. Я в 2022 году ездила чаще на такси, а в 2023-м поездки сильно подорожали, где-то в четыре раза местами. Пришлось сильно сокращать количество поездок, чаще пользоваться общественным транспортом, — заявила раздраженная читательница Анастасия.

Однако есть и те, кто считает возросшие цены на такси оправданными. Горожане напоминают, что покупка и обслуживание автомобиля кратно подорожали за последние пару лет, а значит, и цены на услуги по извозу должны были стать выше.

— Представьте, что такси из условного Академа до центра по утрам будет стоить 200 рублей, сколько человек предпочтет комфортную поездку на такси? «А сколько водителей захотят возить ваши недовольные жопы за 200 рублей?» —спрашивает Александр. — Машины будете ждать по пять часов. Задумайтесь. А еще не забывайте, что автомобили, топливо, обслуживание, страховки дорожают, и подумайте, что ваши 350 рублей (считая грязными, а чистыми — 190) за поездку 35 минут и 10 километров — вообще ничто. А еще водитель хочет заработать, а не только отбить расходы, — пояснил водитель Александр.

Важно отметить, как искусственный интеллект (ИИ) играет ключевую роль в нашем проекте по предсказанию цен на такси. С использованием передовых методов машинного обучения и алгоритмов прогнозирования наше приложение анализирует множество факторов, таких как адреса, расстояния, погодные условия и другие, чтобы предсказать возможные изменения цен в ближайшие 30 минут. Этот инновационный подход, основанный на ИИ, не только повышает точность предсказаний, но и создает более прозрачный и удобный опыт для пользователей при заказе такси.

# ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

## Функционирование такси в городской транспортной системе

Такси играют важную роль в обеспечении мобильности в городских районах и являются значительной частью общественного транспорта. Они предоставляют гибкие и оперативные транспортные услуги, которые способны удовлетворить разнообразные потребности пассажиров, включая поездки на короткие и длинные дистанции, а также экстренные ситуации. Такси способствуют улучшению доступности городских районов, особенно тех, которые не охвачены традиционными маршрутами общественного транспорта.

Исследования показывают, что такси значительно влияют на транспортную систему городов, снижая нагрузку на другие виды общественного транспорта и способствуя снижению заторов на дорогах. Они также играют ключевую роль в обеспечении транспортной доступности для лиц с ограниченной подвижностью, что делает их важным элементом инклюзивного городского транспорта. В частности, такси обеспечивают мобильность в городских районах и играют важную роль в общественном транспорте [1].

Эволюция таксомоторной индустрии также имеет значительное влияние на урбанистическую мобильность. Современные технологии, такие как системы вызова такси через мобильные приложения, улучшили доступность и удобство использования такси для населения. Такие сервисы, как Uber и Lyft, изменили традиционную модель такси, предлагая новые формы транспорта и улучшая качество обслуживания. "Таксомоторная индустрия эволюционировала, интегрируя современные технологии, что существенно улучшило городскую мобильность и доступность" [2].

Таким образом, такси являются неотъемлемой частью городской транспортной системы, обеспечивая гибкость, доступность и оперативность транспортных услуг, что значительно улучшает общее качество городской мобильности.

## Влияние цены на запрос такси

Стоимость поездки на такси является ключевым фактором, влияющим на спрос на эти услуги. Повышение тарифов на такси обычно приводит к снижению спроса, так как пассажиры начинают искать более экономичные альтернативы, такие как общественный транспорт или услуги по вызову автомобилей, например, Uber и Lyft.

Снижение цен, напротив, может стимулировать увеличение числа поездок, так как услуги такси становятся более доступными для широкой аудитории. Исследования показывают, что корректировки цен на такси и связанные с ними факторы существенно влияют на спрос и предлагают целевые политические рекомендации [3]. Например, в условиях повышенной конкуренции с другими видами транспорта, такси должны применять гибкие ценовые стратегии, чтобы оставаться конкурентоспособными и привлекать пассажиров.

Динамическое ценообразование, основанное на текущем спросе и предложении, может стать эффективным инструментом для управления спросом на такси. Такие модели ценообразования помогают сбалансировать нагрузку на транспортную систему, предотвращая перегрузку в часы пик и стимулируя использование такси в периоды низкого спроса.

Однако, чрезмерное повышение тарифов может привести к значительному оттоку пассажиров в пользу более дешевых альтернатив. Это особенно актуально в городах с развитой сетью общественного транспорта и широким распространением услуг по вызову автомобилей. Например, в исследовании, посвященном влиянию стоимости поездки на спрос на такси, отмечается, что пассажиры становятся более чувствительными к изменениям цен в условиях наличия множества альтернативных видов транспорта [3].

Таким образом, для сохранения устойчивого уровня спроса, компании, предоставляющие услуги такси, должны тщательно учитывать ценовую эластичность спроса и адаптировать свои тарифы в соответствии с потребностями и предпочтениями пассажиров. Внедрение инновационных ценовых моделей и постоянный мониторинг рынка помогут поддерживать баланс между стоимостью услуг и уровнем спроса, обеспечивая стабильное функционирование такси в городской транспортной системе.

## Факторы, влияющие на цену запроса такси

### 1.3.1 общие факторы

Цены на услуги такси зависят от множества факторов, включая экономические, технологические и рыночные условия. Экономические факторы, такие как стоимость топлива, амортизация транспортных средств, заработная плата водителей и эксплуатационные расходы, непосредственно влияют на структуру тарифов. Технологические факторы, включая внедрение динамических моделей ценообразования и использование мобильных приложений, также играют важную роль. Рыночная конкуренция с другими видами транспорта, такими как общественный транспорт и услуги по вызову автомобилей, заставляет компании пересматривать и адаптировать свои ценовые стратегии.

Исследование показывает, что динамика спроса и предложения оказывает значительное влияние на цены на такси. В частности, пространственные и временные плотности, часы пик и вторичная плотность дорог являются важными факторами, влияющими на спрос и предложение такси [4]. Например, в условиях высокой плотности движения или в периоды повышенного спроса, таких как часы пик, тарифы на такси могут значительно возрастать.

Динамическое ценообразование, основанное на текущем спросе и предложении, является одним из наиболее эффективных методов управления ценами на такси. Эта система позволяет оперативно корректировать тарифы в зависимости от различных факторов, таких как спрос, погодные условия и цены на топливо [5]. Внедрение таких моделей ценообразования помогает компаниям более эффективно реагировать на изменения рыночных условий и обеспечивать оптимальное распределение транспортных ресурсов.

### 1.3.2 Погодные условия

Погода оказывает значительное влияние на спрос и цены на услуги такси. В неблагоприятных погодных условиях, таких как дождь, снег или сильный ветер, спрос на такси обычно возрастает, что приводит к увеличению тарифов. Исследования показывают, что интенсивность дождя является значимым фактором, влияющим на спрос на такси. По мере увеличения интенсивности осадков, спрос на такси в вечерние часы пик также возрастает [6]. Это связано с тем, что пассажиры предпочитают более комфортные и безопасные условия передвижения, которые обеспечивают такси, особенно в условиях плохой погоды.

Кроме того, погодные условия могут увеличивать время поездки и заторы на дорогах, что также учитывается при динамическом ценообразовании. В таких ситуациях компании часто применяют повышенные тарифы, чтобы компенсировать дополнительные расходы, связанные с увеличением времени в пути и потреблением топлива.

### 1.3.3 Время суток

Время суток является одним из ключевых факторов, влияющих на стоимость поездок на такси. Спрос на такси значительно варьируется в зависимости от времени дня, что отражается на тарифах. В часы пик, когда большинство людей отправляются на работу или возвращаются домой, спрос на такси возрастает, что приводит к увеличению цен. В периоды повышенного спроса, таких как утренние и вечерние часы пик, наблюдаются значительные колебания в спросе на такси, а также в частоте и длительности поездок [7].

Кроме того, водители такси склонны избегать работы в часы пик из-за низких ожидаемых доходов, несмотря на высокий спрос. Водители часто выбирают работать в периоды с более стабильными доходами, что еще больше увеличивает тарифы на такси в часы пик [8].

Ночью, когда общественный транспорт работает в ограниченном режиме, тарифы на такси могут возрастать из-за повышенного спроса и уменьшенного предложения. Динамическое ценообразование позволяет компаниям оперативно реагировать на изменения спроса в течение дня, обеспечивая более справедливые и устойчивые тарифы для пассажиров.

### 1.3.4 Специальные события

Специальные события, такие как концерты, спортивные мероприятия, праздники и фестивали, оказывают значительное влияние на цены на такси. В такие периоды спрос на услуги такси резко возрастает, что приводит к временному увеличению тарифов. Например, исследования показывают, что погодные условия, особенно дождь, существенно влияют на транспортный спрос, подчеркивая, как особые условия, такие как погодные события, могут влиять на использование такси [9]. В периоды проведения массовых мероприятий компании часто вводят специальные тарифы, чтобы справиться с увеличенным спросом и обеспечить достаточное количество транспортных средств для обслуживания пассажиров.

Кроме того, в такие дни часто наблюдается увеличение пробок на дорогах и времени в пути, что также учитывается при определении тарифов. Динамическое ценообразование позволяет компаниям оперативно реагировать на изменения спроса и корректировать тарифы в реальном времени, обеспечивая баланс между спросом и предложением.

Особые события и праздники требуют гибкого подхода к ценообразованию для поддержания устойчивого функционирования таксомоторной индустрии, учитывая их влияние на резкие колебания спроса.

## Способы определения цены такси

Цены на такси могут определяться различными моделями ценообразования, каждая из которых имеет свои особенности и применяется в зависимости от условий рынка и регуляторных требований. Одной из наиболее распространенных моделей является установленная цена за время или расстояние поездки. Различные системы тарифов на такси, включая расчет стоимости на основе времени или расстояния поездки, обсуждаются в контексте влияния конкуренции и регулирования на эти модели [10].

Динамическое ценообразование является еще одной важной моделью, которая позволяет оперативно корректировать тарифы в зависимости от текущего спроса, погодных условий и цен на топливо. Эта система позволяет устанавливать цены в режиме реального времени, что обеспечивает более точное соответствие между спросом и предложением [11]. Динамическое ценообразование играет ключевую роль в адаптации к изменяющимся рыночным условиям и повышении эффективности работы такси.

Кроме того, на стоимость поездок на такси существенно влияют такие факторы, как расстояние, время в пути и дорожные условия. Интеграция зарядов за расстояние и время в расчет тарифов, а также различия между тарифами по счетчику и фиксированными тарифами, играют важную роль в формировании конечной стоимости поездки [12]. Эти факторы учитываются при расчете окончательной стоимости, что обеспечивает прозрачность и справедливость тарифов для пассажиров.

Разнообразие методов определения цены такси, включая как традиционные модели, так и современные подходы, учитывающие динамику рынка и технологические инновации, позволяет разрабатывать более эффективные стратегии ценообразования и обеспечивать устойчивое функционирование таксомоторной индустрии.

# ТЕОРЕТИЧЕСКАЯ ОСНОВА И МЕТОДЫ

## Способы сбора данных

### Основы теории сбора данных

Первичные и вторичные данные:

* Первичные данные: эти данные собираются непосредственно для целей исследования. Примеры включают опросы, эксперименты и наблюдения. Они позволяют получить точную информацию, адаптированную под специфические требования исследования. Например, проведение опросов среди пассажиров такси может дать представление об их предпочтениях и уровне удовлетворенности услугами.
* Вторичные данные: эти данные уже были собраны ранее и используются для новых исследований. Примеры включают данные из баз данных, отчетов или других источников, таких как Yandex Taxi для информации о ценах и тарифах, Weather API для данных о погодных условиях, и 2GIS для данных о расстояниях и дорожных условиях. Вторичные данные экономят время и ресурсы, предоставляя доступ к уже обработанной информации[13].

Количественные и качественные данные:

* Количественные данные: это данные, которые можно измерить и выразить численно. Они включают такие параметры, как цены на такси, расстояния поездок и время в пути. Количественные данные позволяют проводить статистический анализ и выявлять тенденции и зависимости.
* Качественные данные: это описательные данные, которые помогают понять контекст или причины тех или иных явлений. Примеры включают отзывы пользователей, которые могут дать представление о качестве обслуживания и причинах предпочтений пассажиров. Качественные данные дополняют количественные, предоставляя более глубокое понимание исследуемого вопроса[13].

### 2.1.2 Методы сбора данных

Опросы и анкеты: Методика сбора первичных данных, которая позволяет получить конкретную информацию от респондентов. Опросы могут проводиться онлайн, по телефону или лично.

Наблюдения: Сбор данных путем прямого наблюдения за объектами или процессами. Это может включать мониторинг трафика или изучение поведения пассажиров.

Использование API: Важный метод сбора вторичных данных. Например, Yandex Taxi API предоставляет данные о ценах, Weather API предоставляет данные о погоде, а 2GIS — информацию о дорожных условиях и пробках.

Интеграция данных: Слияние данных из различных источников: для получения полной картины важно объединить данные из различных источников. Например, интеграция данных о ценах из Yandex Taxi, погодных данных из Weather API и данных о пробках из 2GIS позволяет провести комплексный анализ факторов, влияющих на стоимость поездок на такси[14].

### 2.1.3 Преимущества и недостатки различных методов

**API**

**Преимущества:**

* Высокая точность данных: Данные, полученные через API, как правило, являются точными и актуальными, так как они поступают непосредственно из источника.
* Обновление данных в реальном времени: API позволяют получать данные в режиме реального времени, что особенно полезно для приложений, требующих постоянного обновления информации, например, для отслеживания цен на такси или текущих погодных условий.
* Автоматизация процесса сбора данных: Использование API позволяет автоматизировать сбор данных, что снижает вероятность ошибок и экономит время, так как исключается необходимость ручного ввода данных.

**Недостатки:**

* Зависимость от сторонних сервисов: Доступность и качество данных зависят от внешних поставщиков услуг. Если API становится недоступным или изменяются его условия использования, это может негативно повлиять на ваш проект.
* Возможные ограничения в использовании данных: Многие API имеют ограничения на количество запросов в определённый период времени или требуют оплаты для получения доступа к полному набору данных. Это может ограничить возможности сбора данных.

**Веб-скрапинг**

**Преимущества:**

* Доступ к данным, которые не предоставляются через API: Веб-скрапинг позволяет извлекать данные с веб-сайтов, которые не предоставляют API, что расширяет возможности для сбора необходимой информации.
* Гибкость в выборе источников данных: Этот метод позволяет извлекать данные из различных источников, что полезно для получения комплексной информации и анализа.

**Недостатки:**

* Юридические и этические проблемы: Веб-скрапинг может нарушать правила использования веб-сайтов или законы о защите данных. Необходимо тщательно проверять легальность и этичность использования этого метода.
* Изменения на сайте могут сломать скрипты: Веб-страницы могут изменяться, что может привести к необходимости постоянного обновления и корректировки скриптов для скрапинга данных.

**Интеграция данных**

**Преимущества:**

* Анализ данных из разных источников: Интеграция данных позволяет объединять информацию из различных источников, что помогает создать более полное представление о ситуации и проводить комплексные анализы.
* Создание комплексных отчетов и анализов: Объединенные данные можно использовать для создания детализированных отчетов и анализа, что полезно для принятия обоснованных решений.

**Недостатки:**

* Сложность реализации: Интеграция данных из различных источников может быть сложной задачей, требующей значительных ресурсов и навыков в области управления данными и программирования.
* Качественное управление данными: Для успешной интеграции данных необходимо обеспечить их правильное управление и очистку, что требует времени и усилий.

**Опросы и анкеты**

**Преимущества:**

* Сбор данных непосредственно от целевой аудитории: Опросы и анкеты позволяют получать информацию прямо от респондентов, что обеспечивает высокую релевантность данных для конкретного исследования.
* Возможность охватить широкий круг респондентов: С помощью онлайн-платформ и других методов можно опросить большое количество людей, что увеличивает репрезентативность выборки.
* Подходят для сбора как количественных, так и качественных данных: Опросы можно использовать для сбора различных типов данных, что позволяет получить как численные показатели, так и детализированные описания и мнения.
* Легкость в автоматизации: Современные онлайн-платформы для проведения опросов позволяют автоматизировать процесс сбора и анализа данных, что экономит время и ресурсы.

**Недостатки:**

* Ошибки и предвзятость в ответах: Респонденты могут предоставлять неточные или предвзятые ответы, что может повлиять на качество данных.
* Не всегда высокий уровень отклика: Участие в опросах добровольное, и не все респонденты готовы тратить время на их заполнение, что может снизить количество собранных данных.
* Требуют тщательного дизайна вопросов: Для получения точных и полезных данных необходимо тщательно продумать формулировку вопросов и структуру анкеты, чтобы минимизировать двусмысленность и ошибки в интерпретации.

## Способы обработки данных

### 2.2.1 Очистка данных

**Очистка данных (Data Cleaning) —** это один из самых важных этапов в процессе обработки данных. Он направлен на обеспечение целостности и надежности данных, что критически важно для точного анализа и предсказаний. Очистка данных включает в себя удаление или исправление ошибок и несоответствий в данных, таких как пропущенные значения, дубликаты, опечатки и выбросы[15].

**Основные цели очистки данных:**

* Повышение качества данных: Убедиться, что данные точны, полны и согласованы.
* Улучшение производительности моделей: Чистые данные обеспечивают более точные и надежные результаты моделирования.
* Снижение риска ошибок: Исправление ошибок и несоответствий уменьшает вероятность получения неверных выводов.

**Популярные методы очистки данных:**

**Удаление пропущенных значений (Missing Value Imputation):**

* Удаление строк: Если количество пропущенных значений невелико, можно удалить строки с пропусками.
* Заполнение средним/медианным/модой: Замена пропущенных значений на среднее, медиану или моду.
* Прогнозирование пропущенных значений: Использование машинного обучения для прогнозирования пропущенных данных на основе других признаков.

**Удаление дубликатов (Duplicate Removal):**

* Обнаружение дубликатов: Поиск строк с идентичными значениями всех или некоторых ключевых признаков.
* Удаление дубликатов: Удаление избыточных записей.

**Обработка выбросов (Outlier Treatment):**

* Обнаружение выбросов: Использование статистических методов, таких как межквартильный размах (IQR) или метод z-оценок.
* Удаление или замена выбросов: Удаление выбросов или замена их значениями, близкими к медиане.

**Исправление опечаток (Error Correction):**

* Обнаружение опечаток: Использование алгоритмов распознавания шаблонов или ручного обзора для выявления ошибок.
* Исправление данных: Автоматическое или ручное исправление обнаруженных ошибок.

**Почему это важно:**

Очистка данных важна, потому что грязные данные могут привести к неверным выводам и моделям, которые не будут хорошо работать в реальных условиях. Например, пропущенные значения могут исказить статистический анализ, а выбросы могут существенно влиять на средние значения и другие статистики.

**Как это делать:**

* Анализ данных: Начните с детального анализа набора данных, чтобы понять, какие проблемы существуют.
* Выбор методов очистки: В зависимости от выявленных проблем, выберите подходящие методы очистки.
* Применение методов: Примените выбранные методы для очистки данных. Например, для обработки пропущенных значений можно использовать метод заполнения средним.
* Проверка данных: После очистки данных необходимо провести проверку, чтобы убедиться, что все проблемы устранены и данные готовы к дальнейшему анализу.

### 2.2.2 Преобразование данных

**Преобразование данных (Data Transformation) —** это процесс преобразования исходных данных в формат, подходящий для анализа и моделирования. Этот этап необходим для того, чтобы обеспечить наилучшие результаты при использовании данных в аналитических моделях и алгоритмах машинного обучения[16].

**Основные цели преобразования данных:**

* Улучшение совместимости: Приведение данных к формату, который совместим с выбранными методами анализа и моделирования.
* Повышение качества моделей: Преобразованные данные могут улучшить производительность моделей за счет создания новых информативных признаков.
* Снижение сложности анализа: Упрощение структуры данных делает анализ более понятным и эффективным.

**Популярные методы преобразования данных:**

1. **Кодирование категориальных признаков (Categorical Encoding):**

* One-Hot Encoding: Преобразование категориальных переменных в набор бинарных переменных.
  + - Label Encoding: Присвоение каждому категориальному значению уникального числового кода.
* Target Encoding: Замена категориальных значений на статистику целевой переменной (например, среднее значение целевой переменной для каждой категории).

1. **Создание новых признаков (Feature Engineering):**

* Полиномиальные признаки (Polynomial Features): Создание новых признаков путем возведения существующих признаков в степень.
* Взаимодействие признаков (Interaction Features): Создание новых признаков путем перемножения существующих признаков.
* Агрегация признаков (Aggregation): Создание новых признаков путем объединения нескольких исходных признаков (например, суммирование или усреднение значений).

1. **Скалирование и нормализация (Scaling and Normalization):**

* Min-Max Scaling: Преобразование данных в диапазон от 0 до 1.
* Standardization (Z-score normalization): Преобразование данных таким образом, чтобы они имели нулевое среднее и единичное стандартное отклонение.
* Robust Scaling: Преобразование данных с использованием медианы и межквартильного размаха для уменьшения влияния выбросов.
* Логарифмическое преобразование (Log Transformation):
* Log Transformation: Преобразование данных путем применения логарифмической функции для уменьшения асимметрии распределения признаков.

**Почему это важно:**

Преобразование данных важно, потому что исходные данные часто находятся в формате, который не подходит для прямого использования в аналитических моделях. Преобразованные данные могут улучшить результаты моделей за счет повышения их устойчивости, уменьшения смещения и повышения точности предсказаний. Например, кодирование категориальных признаков позволяет моделям эффективно работать с нечисловыми данными, а скалирование признаков улучшает сходимость алгоритмов оптимизации.

**Как это делать:**

* Анализ признаков: Изучите каждый признак, чтобы определить, какие преобразования необходимы. Например, категории могут потребовать кодирования, а числовые данные — скалирования.
* Выбор методов преобразования: В зависимости от анализа данных выберите подходящие методы преобразования. Например, для категориальных признаков может подойти One-Hot Encoding.
* Применение преобразований: Примените выбранные методы к данным. Это может включать использование библиотек для машинного обучения, таких как Scikit-Learn, которые предоставляют готовые функции для большинства преобразований.
* Проверка результатов: После преобразования данных проведите проверку, чтобы убедиться, что данные находятся в нужном формате и готовы к дальнейшему анализу.

### 2.2.3 Интеграция данных

**Интеграция данных (Data Integration) —** это процесс объединения данных из различных источников для создания единого, согласованного и полного набора данных. Этот этап критически важен для получения целостного представления о данных и улучшения точности аналитических моделей[17].

**Основные цели интеграции данных:**

* Создание целостного набора данных: Объединение данных из разных источников позволяет получить более полную и точную картину.
* Повышение качества данных: Сведение данных из разных источников может выявить и устранить несоответствия и ошибки.
* Улучшение аналитических возможностей: Объединенные данные предоставляют более богатый контекст для анализа и моделирования, что может привести к более точным и информативным результатам.

**Популярные методы интеграции данных**:

1. Объединение данных (Data Merging):

* Inner Join: Объединение только тех записей, которые имеют соответствующие значения в обоих источниках данных.
* Outer Join: Объединение всех записей из обоих источников данных, при этом недостающие значения заполняются пропусками.
* Left/Right Join: Объединение всех записей из одного источника данных и только соответствующих записей из другого источника.

1. **Объединение данных по столбцам (Concatenation):**

* Vertical Concatenation: Добавление новых строк данных из одного источника в конец данных из другого источника.
* Horizontal Concatenation: Объединение данных путем добавления новых столбцов данных к существующим строкам.
* Согласование данных (Data Reconciliation):
* Сопоставление и выравнивание данных: Устранение несоответствий между данными из разных источников, приведение их к общему формату.
* Агрегация данных: Объединение данных на разных уровнях детализации для получения единого представления.

**Интеграция данных в реальном времени (Real-time Data Integration):**

* ETL-процессы (Extract, Transform, Load): Извлечение данных из источников, их преобразование и загрузка в целевой набор данных.
* Streaming Data Integration: Объединение потоковых данных в реальном времени для создания актуального и динамического набора данных.

**Почему это важно:**

Интеграция данных важна, потому что данные часто хранятся в различных системах и форматах, что затрудняет их использование в анализе. Объединение данных из разных источников позволяет создать более полную и точную картину, что значительно повышает качество аналитических моделей и точность предсказаний. Например, объединение данных о такси с данными о погоде и дорожных условиях может существенно улучшить предсказание стоимости поездки.

**Как это делать:**

* Анализ источников данных: Изучите все доступные источники данных, чтобы понять их структуру, формат и содержимое.
* Планирование интеграции: Определите, какие данные нужно объединить, и выберите подходящие методы интеграции.
* Подготовка данных: Преобразуйте данные в совместимый формат, устраните несоответствия и ошибки.
* Выполнение интеграции: Примените выбранные методы интеграции для объединения данных. Используйте инструменты и технологии, такие как SQL для объединения данных в базах данных или ETL-платформы для автоматизации процесса.
* Проверка и валидация данных: После интеграции проведите проверку, чтобы убедиться в целостности и согласованности данных.

### 2.2.4 Нормализация

**Нормализация (Normalization) —** это процесс преобразования данных таким образом, чтобы они соответствовали определенному диапазону или распределению. Это критически важный шаг в обработке данных, который позволяет обеспечить равное участие всех признаков в анализе и моделировании[18].

**Основные цели нормализации:**

* Снижение влияния масштабов признаков: Приведение признаков к общему масштабу, чтобы ни один из них не доминировал над другими.
* Улучшение сходимости моделей: Нормализованные данные помогают ускорить и улучшить процесс обучения моделей.
* Повышение точности моделей: Уменьшение риска получения искаженных результатов за счет равного вклада всех признаков.

**Популярные методы нормализации данных**:

1. Min-Max Scaling (Мини-максимальное масштабирование):

**Описание**: Преобразование данных в диапазон от 0 до 1.

Применение:

**Преимущества:** Простой и понятный метод, хорошо работает с данными, у которых нет выбросов.

**Standardization (Z-score normalization) (Стандартизация):**

**Описание**: Преобразование данных так, чтобы они имели нулевое среднее значение и единичное стандартное отклонение.

Применение:

**Преимущества**: Хорошо работает с данными, имеющими нормальное распределение.

**Robust Scaling (Устойчивое масштабирование):**

**Описание**: Преобразование данных с использованием медианы и межквартильного размаха (IQR) для уменьшения влияния выбросов.

**Применение**:

**Преимущества**: Менее подвержен влиянию выбросов по сравнению с Min-Max Scaling и стандартизацией.

**Log Transformation (Логарифмическое преобразование):**

**Описание**: Преобразование данных путем применения логарифмической функции для уменьшения асимметрии распределения признаков.

**Применение**:

**Преимущества**: Эффективно для работы с данными, имеющими сильно смещенное распределение.

**Почему это важно:**

* Нормализация данных важна, потому что исходные данные могут иметь различные масштабы и распределения, что может негативно повлиять на обучение моделей машинного обучения. Без нормализации признаки с большими значениями могут доминировать над признаками с меньшими значениями, что приведет к искаженным результатам.
* Нормализованные данные обеспечивают более стабильное и быстрое обучение моделей, а также повышают точность предсказаний.
* Как это делать:
* Анализ данных: Изучите распределение каждого признака, чтобы определить, какой метод нормализации подходит лучше всего.
* Выбор метода нормализации: В зависимости от характера данных и задач анализа выберите подходящий метод нормализации.
* Применение нормализации: Примените выбранный метод к данным. Это можно сделать с помощью библиотек для машинного обучения, таких как Scikit-Learn, которые предоставляют готовые функции для нормализации.
* Проверка результатов: После нормализации данных проведите проверку, чтобы убедиться, что данные масштабированы правильно и готовы к дальнейшему анализу и моделированию.

## Способы защиты модели от переобучения

### 2.3.1 Кросс-валидация

Разделение данных на обучающие и проверочные наборы для обеспечения хорошей работы модели на новых данных

**Кросс-валидация (Cross-Validation) —** это метод оценки производительности модели на невидимых данных. Основная цель кросс-валидации — обеспечить, чтобы модель не только хорошо работала на обучающих данных, но и на новых, ранее невидимых данных[19].

**Основные цели:**

* Обеспечение надежности модели: Проверка, как модель будет работать на новых данных.
* Предотвращение переобучения: Обнаружение ситуаций, когда модель слишком хорошо подстраивается под обучающие данные.

**Популярные методы:**

* K-Fold Cross-Validation: Разделение данных на K равных частей, где модель обучается на K-1 частях и тестируется на оставшейся части. Этот процесс повторяется K раз[19].
* Leave-One-Out Cross-Validation (LOOCV): Особый случай K-Fold Cross-Validation, где K равно количеству наблюдений в данных. Модель обучается на всех данных, кроме одного наблюдения, и тестируется на этом одном наблюдении. Процесс повторяется для каждого наблюдения[19].

**Почему это важно:**

Кросс-валидация важна, потому что она позволяет оценить модель на данных, которые не использовались для ее обучения, что помогает выявить и предотвратить переобучение.

**Как это делать:**

* Разделение данных: Разделите данные на обучающие и тестовые наборы.
* Обучение модели: Обучите модель на обучающем наборе данных.
* Проверка модели: Оцените модель на тестовом наборе данных.
* Повторение процесса: Повторите процесс несколько раз (в случае K-Fold Cross-Validation) для получения надежной оценки производительности модели.

### 2.3.2 Регуляризация

Добавление штрафа за сложность модели для предотвращения переобучения (например, Lasso, Ridge регрессия)

**Регуляризация (Regularization) —** это метод, который добавляет штраф за сложность модели в функцию потерь. Это помогает уменьшить вероятность переобучения, так как модель склоняется к более простым решениям.

**Основные цели:**

* Уменьшение сложности модели: Снижение количества или влияния признаков, чтобы модель не переобучалась.
* Повышение обобщающей способности модели: Обеспечение лучшей производительности модели на новых данных.

**Популярные методы:**

1. Lasso Регрессия (L1 Regularization): Добавление штрафа, равного сумме абсолютных значений коэффициентов.

**Применение**:

Loss Function + λ∑∣ ∣

**Преимущества**: Способствует созданию разреженных моделей, где многие коэффициенты равны нулю.

1. Ridge Регрессия (L2 Regularization): Добавление штрафа, равного сумме квадратов коэффициентов.

**Применение**:

Loss Function + λ∑

**Преимущества**: Снижает величину коэффициентов, но не обнуляет их.

1. Elastic Net Регрессия: Комбинация L1 и L2 регуляризаций.

**Применение**:

Loss Function + ∑ + ∑

**Преимущества**: Сочетает преимущества обоих методов.

**Почему это важно:**

Регуляризация важна, потому что она позволяет избежать переобучения, улучшая обобщающую способность модели. Это достигается за счет штрафов, которые уменьшают влияние менее значимых признаков.

**Как это делать:**

* Выбор метода регуляризации: Определите, какой метод регуляризации подходит для вашей задачи (L1, L2 или Elastic Net).
* Добавление регуляризационного термина: Включите регуляризационный термин в функцию потерь модели.
* Настройка параметра регуляризации: Оптимизируйте параметр регуляризации (𝜆) для достижения баланса между подгонкой и обобщением.
* Обучение модели: Обучите модель с регуляризацией и оцените ее производительность.

### 2.3.3 Обрезка

Снижение сложности деревьев решений путем удаления секций, которые имеют небольшую значимость

**Обрезка (Pruning) —** это метод, используемый для снижения сложности деревьев решений путем удаления ветвей, которые имеют незначительное влияние на результат.

**Основные цели:**

* Уменьшение размера дерева: Удаление ветвей, которые добавляют мало информации, чтобы сделать дерево менее сложным.
* Улучшение обобщающей способности: Снижение переобучения путем упрощения модели.

**Популярные методы:**

* Pre-Pruning (Предварительное обрезание): Обрезка ветвей во время построения дерева на основе определенных критериев, таких как минимальное количество выборок для разделения.
* Post-Pruning (Пост-обрезание): Обрезка ветвей после построения полного дерева, основываясь на оценке значимости ветвей.
* Cost-Complexity Pruning: Удаление ветвей, которые уменьшают общую ошибку на тестовых данных.

**Почему это важно:**

Обрезка важна, потому что деревья решений склонны к переобучению, особенно когда они становятся слишком сложными. Обрезка помогает создать более простую и обобщающую модель, которая лучше работает на новых данных.

**Как это делать:**

* Построение дерева решений: Постройте дерево решений на обучающих данных.
* Оценка ветвей: Оцените значимость каждой ветви на основе ее вклада в общую ошибку.
* Удаление незначимых ветвей: Удалите ветви, которые добавляют мало информации или увеличивают сложность модели без значительного улучшения.
* Проверка и тестирование: Проверьте производительность обрезанного дерева на тестовых данных, чтобы убедиться в улучшении обобщающей способности.

## Алгоритм регрессии

### 2.4.1 Линейная регрессия

Линейная регрессия представляет собой метод статистического анализа, который используется для моделирования зависимости между одной зависимой переменной и одной или несколькими независимыми переменными. Основная идея заключается в нахождении наилучшей прямой (линии регрессии), которая минимизирует сумму квадратов отклонений наблюдаемых значений от предсказанных. Математически модель линейной регрессии описывается уравнением

где Y — зависимая переменная, — независимые переменные,  — свободный член (интерсепт), — коэффициенты регрессии, а 𝜖 — ошибка модели. Ключевым шагом является оценка коэффициентов 𝛽 методом наименьших квадратов, который минимизирует сумму квадратов ошибок (разниц между наблюдаемыми и предсказанными значениями).

Этот метод широко применяется, когда необходимо предсказать значение зависимой переменной на основе известных значений независимых переменных. Линейная регрессия нашла свое применение в таких областях, как экономика, финансы, социология, маркетинговые исследования и многих других[20].

Основные преимущества линейной регрессии включают её простоту и интерпретируемость. Модель легко объяснима и позволяет выявить влияние каждой независимой переменной на зависимую. Однако, несмотря на эти достоинства, линейная регрессия имеет и недостатки: она чувствительна к выбросам и предполагает линейную связь между переменными, что не всегда соответствует реальности.

Линейная регрессия является одним из самых популярных методов анализа данных благодаря своей простоте и эффективности. Она широко используется в академических исследованиях, бизнесе и науке, особенно в таких областях, как эконометрика, биостатистика и социальные науки. Тем не менее, существуют и альтернативные методы, такие как полиномиальная регрессия, логистическая регрессия и методы машинного обучения, включая решающие деревья и случайные леса.

Эти альтернативы лучше использовать в тех случаях, когда данные имеют нелинейные зависимости или содержат сложные взаимосвязи между переменными.

### 2.4.2 Метод наименьших квадратов (МНК) регрессия

Метод наименьших квадратов (МНК) регрессия, также известная как OLS (Ordinary Least Squares) регрессия, представляет собой один из самых распространенных и фундаментальных методов для оценки параметров линейной регрессионной модели. Основная идея МНК заключается в минимизации суммы квадратов ошибок, то есть разницы между наблюдаемыми и предсказанными значениями зависимой переменной. Математически это выражается следующим образом: если есть набор данных с 𝑛 наблюдениями, уравнение линейной регрессии можно записать как

где — значение зависимой переменной для i-го наблюдения, значение p-й независимой переменной для i-го наблюдения, — коэффициенты регрессии, — ошибка модели[21].

Метод наименьших квадратов стремится найти такие значения коэффициентов β , которые минимизируют функцию ошибки:

Это достигается путем решения системы нормальных уравнений, которые выводятся из условий равенства нулю частных производных функции ошибки по каждому коэффициенту.

Основные этапы выполнения МНК регрессии включают сбор и подготовку данных, построение регрессионной модели, расчет коэффициентов регрессии методом наименьших квадратов и интерпретацию полученных результатов. Преимущества МНК регрессии включают её математическую строгость и простоту интерпретации коэффициентов. Однако этот метод также имеет свои ограничения: он предполагает линейность отношений между переменными и чувствителен к выбросам и мультиколлинеарности.

Метод наименьших квадратов широко распространен в анализе данных и применяется в различных областях, таких как экономика, инженерия, медицина и социальные науки. Для успешного применения МНК регрессии важно, чтобы данные удовлетворяли ключевым предположениям модели, таким как нормальность распределения ошибок, гомоскедастичность и отсутствие автокорреляции.

### 2.4.3 Бэггинг (Bootstrap Aggregating)

Бэггинг, или Bootstrap Aggregating, представляет собой мощную технику ансамблевого обучения, которая используется для улучшения стабильности и точности моделей машинного обучения. Основная идея бэггинга заключается в создании множества подвыборок из исходного набора данных с помощью метода бутстрапинга, который предполагает случайное выборочное повторение. Каждая из этих подвыборок используется для обучения отдельной модели, и затем результаты всех моделей объединяются (агрегируются) для получения итогового предсказания[22].

Математически процесс бэггинга можно описать следующим образом: пусть у нас есть исходный набор данных

Методом бутстрапинга создаются k подвыборок каждая из которых содержит nnn наблюдений, выбранных случайным образом с возвращением из исходного набора данных. Для каждой подвыборки обучается своя модель где . Финальное предсказание для нового наблюдения вычисляется как среднее арифметическое предсказаний всех моделей:

для регрессионных задач или с помощью голосования большинства для задач классификации.

Преимущества бэггинга включают уменьшение вариативности модели и повышение её устойчивости к шуму в данных. Это достигается за счет усреднения предсказаний нескольких моделей, что снижает вероятность переобучения и делает итоговую модель более стабильной. Бэггинг особенно эффективен при использовании в сочетании с моделями, которые имеют высокую вариативность, такими как деревья решений.

Метод бэггинга нашел широкое применение в различных областях анализа данных, включая экономику, финансы, биоинформатику и маркетинг. Он часто используется в комбинации с деревьями решений, образуя такие мощные ансамблевые методы, как случайный лес (Random Forest), который представляет собой совокупность деревьев решений, обученных с помощью бэггинга.

### 2.4.4 Алгоритм CatBoost

Алгоритм CatBoost представляет собой современный метод градиентного бустинга, разработанный компанией Яндекс для решения задач регрессии и классификации. Одной из уникальных особенностей CatBoost является его способность эффективно работать с категориальными признаками без необходимости предварительного кодирования. Это достигается благодаря использованию метода кодирования категориальных признаков на основе таргета, что позволяет сохранять важную информацию и улучшать качество модели[23].

Математически CatBoost строится на основе классического метода градиентного бустинга.

Пусть:

— обучающий набор данных, где — вектор признаков, а — целевая переменная. Цель заключается в нахождении модели , которая минимизирует функцию потерь . В процессе градиентного бустинга на каждом шаге строится новое дерево решений , которое обучается на остатках предыдущих предсказаний. Обновление модели происходит по следующей формуле:

где — коэффициент обучения.

CatBoost включает несколько усовершенствований, таких как метод "Ordered Boosting", который предотвращает переобучение, и "Oblivious Trees", обеспечивающие высокую эффективность и быстродействие. Ordered Boosting использует особую схему построения деревьев, при которой каждое дерево строится на подмножестве данных, не использовавшихся для построения предыдущих деревьев. Это позволяет значительно снизить вероятность переобучения и улучшить обобщающую способность модели.

Преимущества CatBoost включают его высокую производительность, способность работать с категориальными данными и устойчивость к переобучению. CatBoost широко используется в различных областях, таких как финансы, маркетинг, здравоохранение и рекомендательные системы. Он показал высокую эффективность в соревнованиях по анализу данных и стал одним из популярных инструментов среди специалистов по машинному обучению.

## оценка модели

### 2.5.1 Метрики точности (Accuracy Metrics)

Метрики точности играют ключевую роль в оценке эффективности моделей машинного обучения, особенно в задачах регрессии. Они предоставляют количественные меры, позволяющие определить, насколько точно модель предсказывает результаты по сравнению с реальными значениями. Основные метрики точности включают среднеквадратическую ошибку (MSE), среднюю абсолютную ошибку (MAE) и среднеквадратическую ошибку корня (RMSE). MSE рассчитывается как среднее значение квадратов разностей между предсказанными и реальными значениями:

где — реальное значение, — предсказанное значение, и n - количество наблюдений. MAE, в свою очередь, представляет собой среднее значение абсолютных разностей между предсказанными и реальными значениями:

Эти метрики важны, так как они позволяют выявить точность модели в различных аспектах, таких как средняя ошибка или отклонение. Они отличаются от метрик ошибок тем, что акцентируются на точности предсказаний, а не на анализе ошибок. Уникальность метрик точности заключается в их способности предоставлять простые и понятные числовые показатели, которые легко интерпретировать и использовать для сравнения различных моделей. В некоторых сценариях, таких как модели с высокими требованиями к точности предсказаний, метрики точности могут быть более полезными, чем другие методы оценки[24].

### 2.5.2 Метрики ошибок (Error Metrics)

Метрики ошибок являются важным инструментом для оценки качества моделей машинного обучения, особенно в задачах регрессии. Они помогают понять, насколько предсказанные значения отклоняются от реальных значений, и оценивают, насколько хорошо модель справляется с поставленной задачей. К наиболее распространенным метрикам ошибок относятся среднеквадратическая ошибка (MSE), средняя абсолютная ошибка (MAE) и корень из среднеквадратической ошибки (RMSE). Среднеквадратическая ошибка (MSE) вычисляется как среднее значение квадратов разностей между предсказанными и реальными значениями:

где — реальное значение, — предсказанное значение, и n - количество наблюдений. Средняя абсолютная ошибка (MAE) измеряет среднее значение абсолютных разностей между предсказанными и реальными значениями:

Корень из среднеквадратической ошибки (RMSE) предоставляет значение корня из MSE, что делает эту метрику более интерпретируемой в тех же единицах измерения, что и исходные данные:

Научная основа использования метрик ошибок заключается в их способности давать количественные оценки отклонений модели, что позволяет более детально анализировать её производительность. В отличие от метрик точности, метрики ошибок фокусируются на размере и распределении ошибок модели, что делает их незаменимыми для анализа качества предсказаний. Эти метрики позволяют сравнивать модели на основе их способности минимизировать ошибки, выявлять переобучение и другие проблемы. Таким образом, метрики ошибок играют ключевую роль в оценке и сравнении различных алгоритмов машинного обучения, обеспечивая всесторонний анализ их эффективности[24].

Метрики ошибок (Error Metrics) и Метрики точности (Accuracy Metrics) являются важными инструментами для оценки качества моделей машинного обучения, но они фокусируются на различных аспектах производительности модели. Метрики ошибок, такие как среднеквадратическая ошибка (MSE), средняя абсолютная ошибка (MAE) и корень из среднеквадратической ошибки (RMSE), оценивают величину и распределение ошибок модели. Они помогают понять, насколько предсказанные значения отклоняются от реальных значений. Например, MSE и RMSE дают квадратичные оценки ошибок, которые акцентируют внимание на больших отклонениях, в то время как MAE предоставляет среднее абсолютное отклонение без квадратичного усиления.

С другой стороны, метрики точности, такие как коэффициент детерминации (R-squared), измеряют степень, в которой предсказанные значения модели совпадают с реальными данными. Коэффициент детерминации показывает долю дисперсии зависимой переменной, объясняемой независимыми переменными модели, и вычисляется как:

где — реальное значение, — предсказанное значение, и — среднее значение реальных данных.

Основное различие между этими типами метрик заключается в их фокусе: метрики точности оценивают, насколько хорошо модель объясняет данные, тогда как метрики ошибок концентрируются на размерах и распределениях ошибок. В сценариях, где критична минимизация ошибок, предпочтение отдается метрикам ошибок. Когда важно объяснить вариацию данных, метрики точности становятся более полезными. Уникальность каждой группы метрик заключается в их способности предоставлять различный взгляд на производительность модели, что позволяет более комплексно оценивать и сравнивать различные алгоритмы машинного обучения.

### 2.5.3 Коэффициент детерминации (R-squared)

Коэффициент детерминации (R-squared) является одной из ключевых метрик для оценки качества модели регрессии. Он измеряет долю дисперсии зависимой переменной, которая объясняется независимыми переменными модели, и таким образом показывает, насколько хорошо модель объясняет вариацию данных. Коэффициент детерминации вычисляется по формуле:

где — реальное значение, — предсказанное значение, и — среднее значение реальных данных. Значение варьируется от 0 до 1, где 1 указывает на идеальную модель, полностью объясняющую всю вариацию зависимой переменной, а 0 означает, что модель не объясняет вариацию данных вообще[24].

Научная основа использования коэффициента детерминации заключается в его способности предоставлять четкое и интуитивное представление о том, насколько хорошо модель подходит к данным. В отличие от других метрик, таких как среднеквадратическая ошибка (MSE) или средняя абсолютная ошибка (MAE), которые фокусируются на величине ошибок, акцентирует внимание на объяснении вариации данных. Однако коэффициент детерминации имеет свои ограничения; например, он может быть завышен при добавлении ненужных переменных в модель, что не всегда указывает на истинное улучшение качества модели.

Уникальность заключается в его способности предлагать простую интерпретацию отношения объясненной вариации к общей вариации, что делает его важным инструментом при сравнении различных моделей и определении их качества. В сценариях, где важно понять, насколько модель объясняет поведение данных, коэффициент детерминации становится особенно полезным. Это делает его незаменимым инструментом для оценки и улучшения моделей машинного обучения.

# ПРАКТИЧЕСКАЯ ЧАСТЬ

## Сбор данных

### 3.1.1 Получение цены на такси

Для точного сбора данных о ценах на такси я использовал API Яндекс. Этот процесс начался с получения токена доступа, что потребовало прямого общения с Яндекс для получения необходимой авторизации. После получения доступа я настроил систему для отправки HTTP-запросов к API Яндекс каждые пять минут. Эти запросы были разработаны для получения текущей стоимости поездки на такси по определенному маршруту от точки А до точки Б. Автоматизируя процесс отправки запросов, я обеспечил непрерывный поток актуальных данных о ценах. Каждый ответ от API предоставлял самую свежую информацию о тарифах, которая затем фиксировалась вместе с временной меткой запроса. Этот метод позволил собрать детализированные и актуальные данные, что является ключевым для анализа тенденций и колебаний цен на такси с течением времени.

function getPrice(test) {

fetch(test, {

headers: {

'Accept': 'application/json'

}

})

.then(response => response.text())

.then(text => JSON.parse(text))

.then(jsonData => jsonData.options[0].price)

.then(price => console.log(price))

}

### 3.1.2 Получение информации о погоде

Для получения данных о погодных условиях я использовал free weather API. Сначала я настроил систему на JavaScript для отправки HTTP-запросов к этому API через регулярные промежутки времени. Эти запросы были разработаны для получения текущей информации о погоде, включая температуру, ощущаемую температуру и общие погодные условия, такие как солнечно, дождливо, облачно и т.д. Автоматизируя процесс отправки запросов, я обеспечил непрерывный поток актуальных данных о погоде. Каждый ответ от API предоставлял детализированную информацию, которая затем фиксировалась вместе с временной меткой запроса. Этот метод позволил собрать полные и точные данные о погодных условиях, что является важным для анализа их влияния на цены на такси и дорожные условия.

function getTemp(test) {

fetch(test, {

headers: {

'Accept': 'application/json'

}

})

.then(response => response.text())

.then(text => JSON.parse(text))

.then(jsonData => jsonData.current.temp\_c)

.then(temp => console.log("temp: ", temp))

}

function getTempf(test) {

fetch(test, {

headers: {

'Accept': 'application/json'

}

})

.then(response => response.text())

.then(text => JSON.parse(text))

.then(jsonData => jsonData.current.feelslike\_c)

.then(temp => console.log("feels like: ", temp))

}

### 3.1.3 Получение информации о дорожной ситуации

Для получения данных о дорожной ситуации был использован веб-скрейпинг. Сначала была настроена система для извлечения данных с веб-сайтов с использованием JavaScript и библиотеки Cheerio. Этот скрипт был разработан для получения информации о расстоянии между точками A и B, а также текущей дорожной обстановке. Автоматизируя процесс скрейпинга, удалось обеспечить непрерывный поток актуальных данных о дорожной ситуации. Каждый извлечённый набор данных предоставлял информацию о текущем состоянии дорог, включая пробки и заторы, что затем фиксировалось вместе с временной меткой. Этот метод позволил собрать детализированные данные о дорожной ситуации, которые являются важными для анализа влияния дорожных условий на цены на такси и общее время поездки.

const puppeteer = require('puppeteer');

(async () => {

const browser = await puppeteer.launch();

const page = await browser.newPage();

await page.goto('https://2gis.ru/ekaterinburg', { waitUntil: 'networkidle2' });

await page.waitForSelector('button[aria-label="Пробки"]');

const trafficValue = await page.$eval('button[aria-label="Пробки"]', button => button.textContent);

await browser.close();

})();

### 3.1.4 Формирование данных

На этапе формирования данных я объединил все предыдущие этапы в один процесс. С использованием JavaScript я создал скрипт, который отправлял запросы к трем API — Yandex API, free weather API и 2GIS API — через регулярные промежутки времени. Этот скрипт собирал информацию о ценах на такси, погодных условиях и дорожной ситуации. Затем все полученные данные консолидировались в одну строку, включающую время, день недели, расстояние, текущую погоду, температуру, ощущаемую температуру, стоимость поездки и дорожную обстановку. Эти данные записывались в набор данных в формате CSV, что обеспечивало удобное хранение и последующую обработку. Автоматизируя этот процесс, я обеспечил систематический сбор и сохранение полных и актуальных данных, необходимых для анализа и построения моделей.

function addEntry(traffic, temp, tempFeeling, price, day, distance, weather) {

const timeCategory = getTimeCategory();

const record = {

traffic: traffic,

temp: temp,

tempFeeling: tempFeeling,

price: price,

day: day,

timeCategory: timeCategory,

distance: distance,

weather: weather

};

csvWriter.writeRecords([record])

.then(() => {

console.log('Record added successfully.');

});

}

## Обработка данных

### 3.3.1 Очистка данных

Для очистки данных я использовал Python и написал скрипт, который проверял каждую строку собранного набора данных. В этом процессе я проверял наличие пропущенных данных и дублирующихся записей. Также были учтены все стандартные процедуры, выполняемые на этом этапе. Скрипт автоматически обрабатывал данные, удаляя или исправляя проблемные записи, что обеспечивало целостность и точность набора данных. Этот этап был критически важен для подготовки данных к последующей обработке и анализу, гарантируя, что они будут чистыми и пригодными для использования в моделях машинного обучения и других аналитических задачах.

data = pd.read\_csv("taxiprice.csv")

data = data.drop\_duplicates()

data = data.dropna()

columns\_to\_convert = ['price', 'traffic', 'temperature', 'tem\_feels\_like']

data[columns\_to\_convert] = data[columns\_to\_convert].astype('float64')

data = pd.get\_dummies(data, columns=['weather', 'day', 'time'])

x, y = data.drop(['price'], axis=1), data['price']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=42)

### 3.2.2 Преобразование данных

На этапе преобразования данных я использовал Python для выполнения всех необходимых трансформаций, чтобы подготовить данные для анализа и моделирования. В первую очередь, я применил метод One-Hot Encoding для преобразования категориальных данных в бинарные значения. Для этого я превратил дни недели в отдельные бинарные столбцы, где каждый день недели представлен отдельной колонкой с значениями 0 и 1, указывающими на отсутствие или присутствие данного дня. Этот метод позволил устранить проблему порядковости в категориальных данных и улучшить качество моделей машинного обучения.

Далее я разделил временные данные на различные категории. Время суток было разделено на утро, день, вечер и ночь. Эти категории также были преобразованы в бинарные значения с помощью One-Hot Encoding, что позволило более точно учитывать влияние времени суток на анализируемые показатели. Например, утренний час пик и вечерние поездки могут значительно различаться по стоимости и времени в пути, и такое преобразование помогает моделям учитывать эти различия.

Для проверки корректности и адекватности выполненных преобразований я использовал тепловую карту (heatmap) для визуализации данных. Тепловая карта показала корреляции между различными категориями данных, что позволило выявить потенциальные проблемы и убедиться в правильности выполненных трансформаций. Визуализация также помогла понять, какие временные интервалы и дни недели наиболее сильно влияют на стоимость поездок и дорожные условия.

Этот этап преобразования данных был критически важен для подготовки данных к последующему анализу и моделированию. Он обеспечил согласованность и сопоставимость данных, что значительно повысило точность и надежность аналитических выводов и моделей машинного обучения.

data['time\_of\_day'] = pd.cut(

pd.to\_datetime(data['time'], format='%H:%M').dt.hour,

bins=[0, 6, 12, 18, 24],

labels=['Night', 'Morning', 'Afternoon', 'Evening'],

include\_lowest=True,

right=False

)

data = pd.get\_dummies(data, columns=['time\_of\_day'], prefix='tod')

plt.figure(figsize=(12, 8))

correlation\_matrix = data.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Heatmap')

plt.show()

x, y = data.drop(['price'], axis=1), data['price']

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=42)

## Выбор алгоритма

### 3.3.1 Реализация алгоритмов

На этапе реализации я разработал и реализовал код для четырех различных алгоритмов машинного обучения:

* линейной регрессии,
* метода наименьших квадратов (МНК) регрессии,
* бэггинга (Bootstrap Aggregating)
* алгоритма CatBoost

Используя Python и библиотеки для машинного обучения, такие как scikit-learn и CatBoost, я написал скрипты для каждого из этих алгоритмов. После этого я провел обучение моделей на собранном и предварительно обработанном наборе данных. Этот процесс включал настройку гиперпараметров и обеспечение правильного обучения моделей, чтобы они могли эффективно анализировать данные и делать прогнозы. Реализация этих алгоритмов позволила создать несколько моделей, которые впоследствии могли быть оценены и сравнены для выбора наилучшего подхода.

# 1. Linear Regression

lr\_model = LinearRegression()

lr\_model.fit(x\_train, y\_train)

y\_pred\_lr = lr\_model.predict(x\_test)

# 2. Ordinary Least Squares (OLS) Regression

x\_train\_ols = sm.add\_constant(x\_train)

ols\_model = sm.OLS(y\_train, x\_train\_ols).fit()

x\_test\_ols = sm.add\_constant(x\_test, has\_constant='add')

y\_pred\_ols = ols\_model.predict(x\_test\_ols)

# 3. Bagging

bagging\_model = BaggingRegressor()

bagging\_model.fit(x\_train, y\_train)

y\_pred\_bagging = bagging\_model.predict(x\_test)

# 4. CatBoost

catboost\_model = CatBoostRegressor(verbose=0)

catboost\_model.fit(x\_train, y\_train)

y\_pred\_catboost = catboost\_model.predict(x\_test)

### 3.3.2 Оценка и выбор алгоритма

После обучения моделей я приступил к их тестированию с использованием тренировочного набора данных. Для оценки производительности каждой модели я использовал коэффициент детерминации (R-squared, R²), который позволяет измерить, насколько хорошо модель объясняет вариативность данных. Сравнивая результаты различных моделей, я смог определить, какая из них показывает наилучшие результаты.

После проведения тестов оказалось, что алгоритм CatBoost продемонстрировал наивысшие значения R², что свидетельствовало о его превосходной точности и способности к обобщению данных. Этот алгоритм значительно превзошел линейную регрессию, метод наименьших квадратов (МНК) регрессию и бэггинг (Bootstrap Aggregating) по всем критериям. На основании этих результатов я выбрал CatBoost в качестве основного алгоритма для дальнейшего использования в проекте, поскольку он показал наилучшие результаты и соответствовал требованиям задачи.

models = {

"Linear Regression": r2\_score(y\_test, y\_pred\_lr),

"OLS Regression": r2\_score(y\_test, y\_pred\_ols),

"Bagging": r2\_score(y\_test, y\_pred\_bagging),

"CatBoost": r2\_score(y\_test, y\_pred\_catboost)

}

best\_model = max(models, key=models.get)

print(f"The best model is: {best\_model} with R2 score: {models[best\_model]}")

## Создание блока fastAPI

Для создания серверного API был использован FastAPI. Этот фреймворк позволяет быстро разрабатывать и развертывать высокопроизводительные API, что делает его идеальным выбором для данного проекта. В первую очередь был настроен серверный проект и определены эндпоинты для получения и обработки данных. Процесс настройки включал в себя определение маршрутов для различных типов запросов, таких как получение текущих данных о дорожной ситуации, погодных условиях и ценах на такси. Также были настроены методы для обработки этих данных и возвращения ответов клиентам.

На следующем этапе были интегрированы ранее разработанные модули сбора данных с FastAPI. Эти модули включали сбор данных о ценах на такси, погодных условиях и дорожной ситуации. Для этого был разработан механизм, который позволял серверному API взаимодействовать с различными источниками данных и агрегировать информацию в реальном времени. Например, данные о дорожной ситуации собирались с помощью веб-скрейпинга, а данные о погоде получались из внешнего API. Серверный API теперь мог принимать запросы, собирать данные из этих источников и формировать строку данных для дальнейшего анализа. Каждый раз, когда поступал запрос, сервер собирал актуальные данные, объединял их и возвращал клиенту в структурированном виде.

Была разработана логика анализа цен на сервере, используя модели машинного обучения, которые были обучены ранее на исторических данных. Эти модели были интегрированы в серверный проект и использовались для прогнозирования цен на такси в зависимости от текущих условий. При поступлении запроса на анализ цены сервер обрабатывал данные и выполнял прогнозирование, определяя, является ли текущая цена высокой или низкой. Для этого использовались различные алгоритмы машинного обучения, такие как линейная регрессия и деревья решений. Результаты прогнозирования возвращались клиенту вместе с рекомендациями, основанными на анализе данных. Этот подход позволил создать мощный инструмент для анализа и прогнозирования цен, который мог бы использоваться в реальном времени для принятия обоснованных решений.

A diagram of a software application

Description automatically generated

## Создание интерфейса React Native

Сначала было установлено окружение для разработки мобильного приложения с использованием React Native. Этот процесс включал в себя установку всех необходимых зависимостей, таких как библиотеки и инструменты для разработки, что обеспечило плавный процесс создания приложения. Были использованы пакеты и инструменты, такие как Node.js, npm, Expo, а также необходимые плагины и расширения для разработки на React Native. Это обеспечило создание надежного и гибкого окружения, способного поддерживать функциональность и производительность приложения.

Для получения актуальных цен на такси был интегрирован готовый блок кода от Яндекс. Этот блок позволял приложению получать информацию о текущих ценах на поездки, предоставляя пользователю возможность вводить маршрут и мгновенно получать актуальную цену. Интеграция включала в себя настройку API Яндекс, обработку данных, получаемых с сервера, и их отображение в интерфейсе приложения. Это обеспечило пользователям точную и своевременную информацию о стоимости поездок, что является ключевым элементом функционала приложения.

Далее был разработан модуль для взаимодействия с сервером FastAPI. Этот модуль был ответственен за отправку запросов на сервер и получение ответов с рекомендациями. Модуль включал в себя методы для формирования запросов, отправки их на сервер, обработки полученных данных и передачи их в приложение для дальнейшего использования. Это взаимодействие обеспечивало бесшовный обмен данными между клиентской и серверной частями приложения, что позволяло получать актуальную информацию в реальном времени и предоставлять ее пользователям.

В приложении была реализована логика рекомендаций на основе полученных данных. Эта логика анализировала текущие и прогнозируемые цены на такси, предоставляя пользователю рекомендации по оптимальному времени заказа такси. Если предсказанная цена была выше текущей, приложение рекомендовало заказать такси немедленно. Если предсказанная цена была ниже текущей, приложение рекомендовало подождать. Такая логика обеспечивала пользователям возможность принимать обоснованные решения, основываясь на анализе данных и прогнозах.

Кроме того, был реализован функционал для прогнозирования цен на следующие 5, 15 и 30 минут. Этот функционал позволял пользователям видеть, как изменятся цены в ближайшем будущем, и принимать более обоснованные решения о том, когда заказывать такси. Прогнозирование базировалось на анализе исторических данных и текущих тенденций, что позволяло предоставлять пользователям точные и надежные прогнозы. Это улучшило пользовательский опыт, делая приложение более полезным и информативным.

Объединение всех этих элементов в одно целое обеспечило создание мощного, функционального и удобного мобильного приложения, способного предоставлять пользователям актуальную информацию о ценах на такси и помогать им принимать оптимальные решения о заказе поездок.

A screenshot of a phone

Description automatically generated

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной выпускной квалификационной работы было разработано приложение для прогноза стоимости на такси, основанное на множественной линейной регрессии. Целью работы было создание инструмента, который бы позволил пользователям предсказывать возможные изменения цен на такси, основываясь на различных факторах в реальном времени, таких как погода, время суток и дорожная обстановка.

Для достижения этой цели был проведен сбор данных с использованием API Yandex Go, OpenWeatherAPI и 2GIS. Собранные данные были сохранены в формате CSV и обработаны с помощью библиотеки scikit-learn в Python. На основе обработанных данных была разработана модель, которая очень точно оценивает изменения цен на такси с использованием метрик средней абсолютной ошибки (MAE) и средней квадратической ошибки (MSE).

В процессе работы мы тщательно изучили различные аспекты функционирования такси, а также факторы, влияющие на их стоимость. Были рассмотрены экономические, погодные и временные факторы, а также влияние дорожной обстановки. Этот анализ позволил нам создать более точную и надежную модель прогнозирования.

Важным этапом работы стало создание серверного API с использованием FastAPI, который позволяет принимать запросы и возвращать предсказания цен. Для создания интерфейса пользователя было использовано React Native, что обеспечивает удобство использования приложения на мобильных устройствах. Приложение позволяет пользователям вводить маршрут и получать актуальную цену поездки, а также прогнозировать изменения цен на следующие 5, 15 и 30 минут, что позволяет принимать обоснованные решения о времени заказа такси.

Разработанное приложение показало высокую точность прогнозов и удобство использования, что делает его полезным инструментом для пользователей, желающих оптимизировать расходы на такси. В будущем планируется дальнейшее улучшение модели путем включения дополнительных факторов и использования более сложных алгоритмов машинного обучения.

Выпускная квалификационная работа закончена «13» \_\_\_\_мая\_\_\_\_ 2024 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

# СПИСОК ИСТОЧНИКОВ

1. Cai H. et al. Understanding taxi travel patterns //Physica A: Statistical Mechanics and its applications. – 2016. – Т. 457. – С. 590-597.
2. Cooper J., Mundy R. Taxi! Urban economies and the social and transport impacts of the taxicab. – Routledge, 2016.
3. Wu, X., Li, J. (2015). Impacts of Travel Cost on Taxi Transport Demand and its Policy Alternatives. In: Zhang, Z., Shen, Z., Zhang, J., Zhang, R. (eds) LISS 2014. Springer, Berlin, Heidelberg.
4. Zhao, Y., & Zhang, Y. (2024). What Factors Affect Taxi Supply and Demand? Transportation Research Part C: Emerging Technologies.
5. Patel, R., & Shah, P. (2023). Dynamic Pricing for Taxi Fares Based on Demand, Weather, and Fuel Prices. Journal of Engineering and Technology Research, 12(5), 738-748.
6. Chen, X., & Wang, Y. (2020). Quantifying the Impact of Rainfall on Taxi Hailing and Operation. Journal of Advanced Transportation.
7. Li, R., & Kamargianni, M. (2018). Hailing in the Rain: Temporal and Weather-Related Variations in Taxi Ridership and Taxi Demand-Supply Equilibrium. Transportation Research Part A: Policy and Practice, 110, 20-32.
8. Wang, H., & Chen, X. (2022). Uncovering Factors Affecting Taxi Income from GPS Traces. Journal of Advanced Transportation, 55(8), 200-215.
9. Liu, X., & Wei, Y. (2017). The Impact of Rainfall on the Temporal and Spatial Distribution of Taxi. Journal of Transport Geography, 62, 77-89.
10. OECD. (2015). Taxi Services Regulation and Competition. OECD Publishing.
11. Patel, R., & Shah, P. (2023). Dynamic Pricing for Taxi Fares Based on Demand, Weather, and Fuel Prices. Journal of Engineering and Technology Research, 12(5), 738-748.
12. LADOT. (2020). LADOT Taxi and For-Hire Vehicle Study. Los Angeles Department of Transportation.
13. Введение зависимостей в .NET. : сайт. // Learn Microsoft. – США. – URL: https://learn.microsoft.com/ru-ru/dotnet/core/extensions/dependency-injection (дата обращения: 06.04.2024).
14. Вязилов, Е.Д. Развитие средств адаптации к экстремальным явлениям, связанным с изменениями климата. : сайт. // Фундаментальная и прикладная климатология. – Россия. – URL: https://fa-climatology.ru/index.php/fpk/article/download/16/27 (дата обращения: 17.06.2024).
15. Дивеев, Р.И., Зубков, О.В. Современные проблемы очистки данных. : сайт. // Инновационное развитие российской науки. – Россия. – URL: http://www.iatp.am/vahanyan/articles/irri2016-informatics.pdf#page=138 (дата обращения: 17.06.2024).
16. Малышев, В.В., Сливкин, С.С. Применение методов машинного обучения для построения рекомендательной системы отбора анкет абитуриентов. : сайт. // КиберЛенинка. – Россия. – URL: https://cyberleninka.ru/article/n/primenenie-metodov-mashinnogo-obucheniya-dlya-postroeniya-rekomendatelnoy-sistemy-otbora-anket-abiturientov (дата обращения: 17.06.2024).
17. Фролова М.С., Фролов С.В., Толстухин И.А. Интеграция медицинской техники в информационную систему лечебно-профилактического учреждения. : сайт. // Вернадский ТГТУ. – Россия. – URL: http://vernadsky.tstu.ru/pdf/2014/03/09.pdf (дата обращения: 17.06.2024).
18. Saranya C., Manikandan G. A study on normalization techniques for privacy preserving data mining. International Journal of Engineering and Technology, 2013. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=35a87b51f7441a87adee91e12eb4d22cd2565556 (дата обращения: 17.06.2024).
19. Jabbar, H., Khan, R.Z. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). : сайт. // academia.edu. – США. – URL: https://www.academia.edu/download/37902828/017.pdf (дата обращения: 17.06.2024).
20. Seber, G.A.F., Lee, A.J. Linear Regression Analysis. : книга. // Wiley Series in Probability and Statistics. – США. – URL: https://books.google.com/books?hl=en&lr=&id=X2Y6OkXl8ysC&oi=fnd&pg=PR5&dq=books+on+linear+regression+models&ots=sfiVD6oMiu&sig=tarPTKlnf1fqQ8CY1Q-NHUGAWYs (дата обращения: 17.06.2024).
21. John, O.O., Nduka, E.C. Quantile regression analysis as a robust alternative to ordinary least squares. : сайт. // Scientia Africana. – researchgate.net. – URL: https://www.researchgate.net/profile/Onyedikachi-John/publication/333403546\_QUANTILE\_REGRESSION\_ANALYSIS\_AS\_A\_ROBUST\_ALTERNATIVE\_TO\_ORDINARY\_LEAST\_SQUARES/links/5cec102f299bf109da74f44a/QUANTILE-REGRESSION-ANALYSIS-AS-A-ROBUST-ALTERNATIVE-TO-ORDINARY-LEAST-SQUARES.pdf (дата обращения: 17.06.2024).
22. Farid, D.M., Rahman, M.Z., Rahman, C.M. An ensemble approach to classifier construction based on bootstrap aggregation. : сайт. // International Journal of Computer Applications. – Citeseer. – URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=11b9f20b8ad759225c9151defbec72b4caa7166c (дата обращения: 17.06.2024).
23. Hussain, S., Mustafa, M.W., Jumani, T.A., Baloch, S.K. A novel feature engineered-CatBoost-based supervised machine learning framework for electricity theft detection. : сайт. // Energy Reports, 2021. – URL: https://www.sciencedirect.com/science/article/pii/S2352484721004716 (дата обращения: 17.06.2024).
24. Raschka, S., Mirjalili, V. Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-learn, and TensorFlow 2. : книга. // Packt Publishing, 2019. – URL: https://books.google.com/books?hl=en&lr=&id=sKXIDwAAQBAJ&oi=fnd&pg=PP1&dq=accuracy+and+error+metrics+in+machine+learning+books&ots=VaDprXWHIl&sig=wdFbChUGYKpPKEyyt\_EB1cdns2g (дата обращения: 17.06.2024).
25. Алферьева Т. И. Методические рекомендации для подготовки и оформления выпускных квалификационных работ: методические рекомендации для обучающихся всех форм обучения укрупненной группы направлений подготовки 09.00.00 Информатика и вычислительная техника / Т. И. Алферьева, В. Н. Васина, Д. Б. Шадрин. – Екатеринбург : УрФУ, 2019. – 78 с. – URL: https://study.urfu.ru/Aid/ViewMeta/13865 (дата обращения: 05.05.2024)