

GeekBrains

**Машинное обучения в задачах прогнозирования**

Программа: Разработчик

Специализация: Data Engineer

ФИО: Алексеева Ксения Олеговна

Санкт-Петербург

2024 г.

Введение.....	3
Глава 1. История и теоретические основы машинного обучения.....	7
1.1 История машинного обучения.....	7
1.2 Области применения машинного обучение .....	11
1.3 Этические аспекты машинного обучения .....	21
1.4 Теоретические основы машинного обучения .....	26
Глава 2. Практическая реализация и анализ результатов .....	33
2.1 Примеры задач машинного обучения на языке Python.....	33
2.2 Решение задачи логистической регрессии на реальном датасете.....	40
Решение задачи.....	40
Заключение .....	43
Список литературы .....	46
Приложение 1 .....	47

## **Введение**

В эпоху стремительного развития технологий и цифровизации всех сфер жизни, машинное обучение становится всё более актуальным и востребованным инструментом для решения сложных задач. Оно позволяет компьютерам обучаться на основе данных и опыта, что делает их способными к самостоятельному анализу и принятию решений. В этом эссе мы рассмотрим актуальность машинного обучения, его применение в различных областях и перспективы развития.

## **Что такое машинное обучение?**

Машинное обучение — это область искусственного интеллекта, которая занимается разработкой алгоритмов и методов, позволяющих компьютерам учиться на основе данных. Алгоритмы машинного обучения анализируют данные, находят закономерности и делают прогнозы или принимают решения. Машинное обучение используется в таких областях, как медицина, финансы, маркетинг, транспорт и другие.

## **Актуальность машинного обучения**

Актуальность машинного обучения обусловлена несколькими факторами:

- **Рост объёмов данных.** С развитием интернета и мобильных устройств, количество данных, доступных для анализа, растёт экспоненциально. Машинное обучение позволяет обрабатывать эти данные и извлекать из них ценную информацию.
- **Сложность задач.** Многие задачи, которые ранее решались вручную, теперь могут быть автоматизированы с помощью машинного обучения. Это позволяет повысить эффективность и производительность труда.
- **Развитие технологий.** Развитие технологий, таких как глубокое обучение, нейронные сети и большие данные, делает машинное обучение более мощным и эффективным.

- **Автоматизация рутинных задач:** Машинное обучение позволяет автоматизировать рутинные задачи, такие как анализ данных, прогнозирование и принятие решений. Это освобождает время человека для более сложных и творческих задач.
- **Повышение производительности:** Машинное обучение может повысить производительность труда за счёт оптимизации процессов и снижения ошибок. Например, в промышленности оно может использоваться для автоматизации производства, контроля качества продукции и предотвращения поломок оборудования.
- **Улучшение качества услуг:** В сфере услуг машинное обучение может быть использовано для персонализации предложений, повышения удовлетворённости клиентов и улучшения качества обслуживания. Например, в здравоохранении оно может помочь врачам в диагностике и лечении заболеваний.
- **Прогнозирование и предотвращение проблем:** Машинное обучение способно прогнозировать возможные проблемы и предлагать меры по их предотвращению. Это может быть полезно в таких областях, как транспорт, где оно может предсказать аварии и предложить альтернативные маршруты.
- **Развитие новых технологий:** Машинное обучение является основой для развития новых технологий, таких как искусственный интеллект и интернет вещей. Оно позволяет создавать более сложные и интеллектуальные системы, способные решать более широкий спектр задач.
- **Адаптация к изменяющимся условиям:** Машинное обучение обладает способностью адаптироваться к изменяющимся условиям и новым данным. Это делает его полезным инструментом для решения сложных задач, требующих постоянного анализа и обновления информации.

- **Снижение затрат:** Машинное обучение может снизить затраты на выполнение определённых задач за счёт автоматизации процессов и повышения эффективности. Это особенно актуально в таких отраслях, как финансы и логистика, где точность и скорость имеют решающее значение.

В целом, актуальность машинного обучения обусловлена его потенциалом для улучшения жизни людей и повышения производительности труда. Оно продолжает развиваться и находить новые применения в различных областях человеческой деятельности.

Машинное обучение находит применение во многих областях, где требуется анализ данных и принятие решений. Вот некоторые примеры:

1. **Медицина:** Машинное обучение может использоваться для диагностики заболеваний, прогнозирования результатов лечения и разработки новых лекарств.
2. **Финансы:** Машинное обучение применяется для анализа финансовых данных, прогнозирования цен на акции и выявления мошенничества.
3. **Маркетинг:** Машинное обучение помогает оптимизировать рекламные кампании, предсказывать спрос на товары и услуги и анализировать поведение потребителей.
4. **Безопасность:** Машинное обучение используется для обнаружения киберугроз, анализа поведения пользователей и предотвращения мошенничества.
5. **Транспорт:** Машинное обучение позволяет оптимизировать маршруты, прогнозировать пробки и обеспечивать безопасность дорожного движения.
6. **Образование:** Машинное обучение может помочь в разработке персонализированных образовательных программ, оценке успеваемости учащихся и прогнозировании результатов экзаменов.
7. **Сельское хозяйство:** Машинное обучение можно использовать для мониторинга состояния посевов, прогнозирования урожайности и оптимизации сельскохозяйственных процессов.

8. **Промышленность.** Машинное обучение может помочь в автоматизации производства, повышении качества продукции и снижении затрат.
9. **Энергетика.** Машинное обучение может быть использовано для прогнозирования спроса на электроэнергию, оптимизации работы электростанций и повышения эффективности энергопотребления.
10. **Экология.** Машинное обучение может применяться для мониторинга окружающей среды

# **Глава 1. История и теоретические основы машинного обучения**

## **1.1 История машинного обучения**

### **История машинного обучения: от первых экспериментов до современных достижений**

Машинное обучение — это область искусственного интеллекта, которая позволяет компьютерам учиться и адаптироваться на основе данных. Оно является одним из самых важных и перспективных направлений в современной науке и технике. В этой статье мы рассмотрим историю развития машинного обучения, его основные этапы и достижения, а также перспективы на будущее.

#### **Первые эксперименты**

Первые попытки создания систем машинного обучения были предприняты ещё в 1950-х годах. Они основывались на принципах статистического анализа и теории вероятностей. Одним из первых примеров машинного обучения была система «Перцептрон», разработанная Фрэнком Розенблаттом в 1958 году. Она представляла собой простую модель нейронной сети, способную распознавать простые геометрические фигуры.

В 1960-е годы были разработаны более сложные модели машинного обучения, такие как деревья решений и логистическая регрессия. Эти методы позволяли решать более сложные задачи, такие как классификация и прогнозирование. Однако они были ограничены в своих возможностях и требовали большого количества данных для обучения.

#### **Развитие методов машинного обучения**

В 1970-е годы началось активное развитие методов машинного обучения. Были разработаны новые алгоритмы, такие как метод опорных векторов, который позволял решать задачи классификации с высокой точностью. Также были

созданы первые системы искусственного интеллекта, способные решать сложные задачи, такие как игра в шахматы и доказательство теорем.

Однако эти системы были очень сложными и дорогостоящими, и их применение было ограничено. В 1980-е годы начался новый этап развития машинного обучения, связанный с появлением новых алгоритмов и методов. Были разработаны методы обучения с подкреплением, которые позволяли машинам самостоятельно обучаться на основе опыта. Также были созданы новые методы кластеризации и классификации, которые позволили решать более сложные и разнообразные задачи.

### **Современные достижения**

В последние десятилетия машинное обучение стало одной из самых быстроразвивающихся областей искусственного интеллекта. Это связано с несколькими факторами, такими как доступность больших объёмов данных, развитие вычислительных мощностей и появление новых методов и алгоритмов.

Одним из наиболее важных достижений последних лет является глубокое обучение, которое представляет собой разновидность машинного обучения, основанную на нейронных сетях. Глубокое обучение позволяет создавать системы, способные выполнять сложные задачи, такие как распознавание речи, изображений и текста. Оно также используется в таких областях, как медицина, финансы и транспорт.

Ещё одним важным достижением является использование машинного обучения в сочетании с другими технологиями, такими как интернет вещей (IoT) и большие данные (Big Data). Это позволяет создавать системы, способные анализировать огромные объёмы данных и принимать решения на основе полученных результатов.

Одним из примеров такого сочетания является система умного дома, которая использует датчики и камеры для сбора информации о состоянии дома и его



обитателях. Затем эта информация анализируется с помощью алгоритмов машинного обучения, чтобы определить, какие действия необходимо предпринять, например, включить свет, открыть дверь или вызвать помощь.

Также машинное обучение используется в таких областях, как медицина, где оно помогает врачам диагностировать заболевания, прогнозировать их течение и назначать лечение. В медицине машинное обучение применяется для анализа медицинских изображений, таких как рентгеновские снимки и МРТ, а также для обработки данных о пациентах, включая историю болезни, результаты анализов и другие показатели.

В последние годы машинное обучение также стало широко использоваться в области искусственного интеллекта (AI), где оно играет ключевую роль в создании систем, способных выполнять сложные задачи, такие как распознавание речи, изображений и текста.

### **Перспективы на будущее**

Машинное обучение продолжает развиваться быстрыми темпами, и перспективы на будущее выглядят очень многообещающими. Ожидается, что в ближайшие годы мы увидим ещё более мощные и эффективные системы машинного обучения, которые будут способны решать ещё более сложные и разнообразные задачи.

Одной из основных тенденций в развитии машинного обучения является увеличение использования больших данных и облачных вычислений. Это позволит создавать более точные и надёжные модели, а также ускорить процесс обучения. Также ожидается, что машинное обучение будет всё больше интегрироваться с другими технологиями, такими как искусственный интеллект и интернет вещей, что приведёт к созданию ещё более мощных и эффективных систем.

Однако развитие машинного обучения также вызывает некоторые опасения. Одним из них является возможность использования машинного обучения для создания систем, которые могут быть использованы для нарушения прав человека или нанесения вреда обществу. Поэтому важно, чтобы разработчики и исследователи машинного обучения учитывали этические аспекты своей работы и стремились создать системы, которые будут приносить пользу людям, а не вред.

## **1.2 Области применения машинного обучение**

Машинное обучение — это область искусственного интеллекта, которая позволяет компьютерам учиться и адаптироваться на основе данных. Оно находит применение во многих сферах человеческой деятельности, от медицины и финансов до транспорта и сельского хозяйства. В этой статье мы рассмотрим несколько примеров того, как машинное обучение используется в разных областях.

### **1. Медицина**

В медицине машинное обучение применяется для анализа медицинских изображений, таких как рентгеновские снимки и МРТ, а также для обработки данных о пациентах, включая историю болезни, результаты анализов и другие показатели. С помощью машинного обучения можно выявить ранние признаки заболеваний сердца или лёгких, что позволит своевременно начать лечение и предотвратить развитие серьёзных осложнений. Также машинное обучение может помочь в разработке новых лекарств и методов лечения, основываясь на анализе больших объёмов данных о биологических процессах и генетических особенностях пациентов.

Например, компания Google разработала систему искусственного интеллекта под названием DeepMind Health, которая помогает врачам диагностировать заболевания и назначать лечение. Система анализирует данные о пациентах из электронных медицинских карт и предлагает рекомендации по лечению. Это позволяет врачам принимать более обоснованные решения и повышать качество медицинской помощи.

### **2. Финансы**

В финансах машинное обучение используется для прогнозирования финансовых рынков, выявления мошенничества и управления рисками. Например, с помощью алгоритмов машинного обучения можно анализировать данные о

ценах на акции, облигации и другие финансовые инструменты, чтобы определить тенденции и предсказать возможные изменения. Это позволяет инвесторам принимать обоснованные решения о покупке и продаже активов, а также минимизировать риски потери капитала.

Также алгоритмы машинного обучения могут использоваться для обнаружения подозрительной активности на счетах клиентов банков, что помогает предотвратить мошенничество и защитить интересы клиентов.

### **3. Транспорт**

В транспорте машинное обучение позволяет оптимизировать транспортные потоки, предсказывать спрос на перевозки и предотвращать аварии. Например, алгоритмы машинного обучения могут анализировать данные о загруженности дорог, скорости движения транспорта и других факторах, чтобы предложить оптимальные маршруты и расписания движения. Это помогает снизить время ожидания пассажиров и улучшить качество обслуживания.

Кроме того, машинное обучение может использоваться для разработки систем автоматического управления транспортом, таких как беспилотные автомобили и поезда. Эти системы могут повысить безопасность дорожного движения и снизить количество аварий.

### **4. Промышленность**

В промышленности машинное обучение может быть использовано для автоматизации производства, контроля качества продукции и предотвращения поломок оборудования. Например, промышленные роботы, оснащённые датчиками и алгоритмами машинного обучения, могут самостоятельно выполнять сложные операции, такие как сварка, сборка и упаковка. Это повышает производительность труда и снижает вероятность ошибок, связанных с человеческим фактором.

Также машинное обучение может использоваться для мониторинга состояния оборудования и выявления потенциальных проблем до того, как они приведут к поломке. Это позволяет предотвратить простои и снизить затраты на ремонт.

Например, компания Siemens использует машинное обучение для анализа данных о работе своих промышленных установок. Система машинного обучения выявляет аномалии в работе оборудования и предлагает рекомендации по их устранению. Это помогает повысить надёжность и эффективность работы установок, а также снизить риск аварий.

## **5. Сельское хозяйство**

В сельском хозяйстве машинное обучение помогает фермерам прогнозировать урожай, определять оптимальные условия выращивания и бороться с вредителями. Например, датчики, установленные на полях, могут собирать данные о влажности почвы, температуре воздуха и других параметрах, которые влияют на рост растений. Алгоритмы машинного обучения анализируют эти данные и выдают рекомендации по поливу, удобрению и защите от вредителей. Это позволяет повысить урожайность и снизить затраты на производство.

Кроме того, машинное обучение может использоваться для разработки систем автоматического управления сельскохозяйственной техникой, таких как тракторы и комбайны. Эти системы могут повысить точность и скорость выполнения работ, а также уменьшить расход топлива.

## **6. Образование**

В образовании машинное обучение используется для персонализации обучения, оценки успеваемости студентов и определения их потребностей в дополнительной помощи. Например, системы машинного обучения могут отслеживать активность студентов во время онлайн-обучения, анализировать их ответы на вопросы и предлагать индивидуальные задания и материалы для

изучения. Это позволяет адаптировать учебный процесс под потребности каждого студента и повысить эффективность обучения.

Также алгоритмы машинного обучения могут использоваться для автоматического оценивания работ студентов, что позволяет сократить время преподавателей на проверку заданий и сосредоточиться на более сложных аспектах обучения.

Например, платформа Knewton использует машинное обучение для адаптации учебных материалов под уровень знаний и интересы каждого студента. Система анализирует результаты тестов и заданий, выполненных студентом, и предлагает ему дополнительные материалы для изучения или повторения. Это позволяет студентам быстрее усваивать материал и достигать лучших результатов.

Кроме того, машинное обучение может использоваться для разработки систем автоматического управления учебным процессом, таких как системы планирования занятий и расписания. Эти системы могут повысить эффективность работы преподавателей и администраторов, а также улучшить качество образования.

## **7. Безопасность**

В области безопасности машинное обучение применяется для обнаружения и предотвращения преступлений, а также для обеспечения безопасности граждан. Например, алгоритмы машинного обучения используются для анализа данных с камер видеонаблюдения, чтобы выявить подозрительное поведение и предупредить правоохранительные органы. Также машинное обучение может быть использовано для разработки систем распознавания лиц и номеров автомобилей, которые помогают в поиске преступников и предотвращении краж.

Компания Amazon разработала систему машинного обучения под названием Rekognition, которая позволяет пользователям загружать фотографии и видео, а система автоматически распознаёт лица и объекты на них. Это может быть

полезно для поиска пропавших людей или для идентификации преступников. Однако эта технология также вызывает опасения по поводу конфиденциальности и прав человека, поэтому её использование должно быть ограничено и контролируемо.

## **8. Спорт**

Машинное обучение находит применение и в спорте. Алгоритмы машинного обучения могут анализировать данные о спортсменах, такие как их физическая подготовка, техника выполнения упражнений и результаты соревнований, чтобы предсказать их будущие выступления и разработать оптимальные стратегии тренировок. Также машинное обучение может помочь тренерам и спортивным аналитикам в анализе данных о командах и игроках, чтобы определить слабые и сильные стороны соперников и разработать тактику игры.

Например, компания IBM разработала систему машинного обучения для анализа данных о баскетболистах, чтобы предсказывать их результативность и вероятность травм. Эта система может быть полезна для тренеров, которые хотят оптимизировать тренировки своих игроков и предотвратить возможные травмы.

Также алгоритмы машинного обучения используются для разработки систем автоматического анализа спортивных событий, таких как определение офсайдов и нарушений правил в футболе или подсчёт очков в теннисе. Эти системы могут повысить точность и объективность судейства, а также ускорить процесс принятия решений.

## **9. Маркетинг и реклама**

В маркетинге и рекламе машинное обучение используется для персонализации предложений и рекомендаций, основанных на интересах и поведении потребителей. Например, алгоритмы машинного обучения анализируют данные о покупках, поисковых запросах и других действиях пользователей, чтобы

предложить им наиболее релевантные товары и услуги. Это позволяет повысить эффективность рекламы и увеличить продажи.

Кроме того, машинное обучение может использоваться для прогнозирования спроса на товары и услуги, что помогает компаниям планировать свои запасы и ценообразование. Также алгоритмы машинного обучения применяются для оптимизации рекламных кампаний, определения наиболее эффективных каналов и форматов рекламы.

Компания Amazon использует машинное обучение для рекомендации товаров покупателям на основе их предыдущих покупок и отзывов. Система анализирует данные о товарах, которые покупатель просматривал или покупал ранее, и предлагает ему похожие товары или новинки. Это помогает Amazon повысить продажи и лояльность клиентов.

## **10. Управление персоналом**

В управлении персоналом машинное обучение применяется для автоматизации процессов найма, оценки и развития сотрудников. Например, алгоритмы машинного обучения могут анализировать резюме и анкеты кандидатов, чтобы выявить наиболее подходящих кандидатов на открытые позиции. Также машинное обучение может быть использовано для оценки эффективности работы сотрудников и определения их потребностей в обучении и развитии.

Система машинного обучения может анализировать данные о производительности, посещаемости, опозданиях и других показателях работы сотрудника, чтобы оценить его эффективность. На основе этой оценки система может предложить сотруднику дополнительные тренинги, курсы или другие возможности для развития. Это помогает сотрудникам повышать

## **10. Управление персоналом**



В управлении персоналом машинное обучение применяется для автоматизации процессов найма, оценки и развития сотрудников. Например, алгоритмы машинного обучения могут анализировать резюме и анкеты кандидатов, чтобы выявить наиболее подходящих кандидатов на открытые позиции. Также машинное обучение может быть использовано для оценки эффективности работы сотрудников и определения их потребностей в обучении и развитии.

Система машинного обучения может анализировать данные о производительности, посещаемости, опозданиях и других показателях работы сотрудника, чтобы оценить его эффективность. На основе этой оценки система может предложить сотруднику дополнительные тренинги, курсы или другие возможности для развития. Это помогает сотрудникам повышать свою квалификацию и производительность, а компании — улучшать качество своей работы и достигать лучших результатов.

Кроме того, машинное обучение используется для прогнозирования текучести кадров и выявления факторов, влияющих на удовлетворённость сотрудников работой. Это позволяет компаниям принимать меры по улучшению условий труда и повышению мотивации персонала.

Например, компания Google разработала систему машинного обучения под названием People Analytics, которая анализирует данные о сотрудниках, такие как их производительность, удовлетворённость работой и вовлечённость, чтобы определить тенденции и предложить рекомендации по управлению персоналом. Эта система помогает Google оптимизировать процессы найма, обучения и развития сотрудников, а также повысить их удовлетворённость и лояльность.

## **11. Здравоохранение**

Машинное обучение находит применение и в здравоохранении. Алгоритмы машинного обучения используются для анализа медицинских данных, таких как результаты анализов, истории болезни и записи о посещениях врачей, чтобы

выявлять закономерности и предсказывать возможные заболевания. Также машинное обучение помогает в разработке новых методов лечения и диагностики, основанных на анализе больших объёмов данных.

Компания IBM разработала систему искусственного интеллекта под названием Watson Health, которая помогает врачам в диагностике и лечении заболеваний. Система анализирует медицинские данные пациента, включая результаты анализов и историю болезни, и предлагает рекомендации по лечению. Это позволяет врачам принимать более обоснованные решения и повышать качество медицинской помощи.

Также алгоритмы машинного обучения применяются для оптимизации работы медицинских учреждений, таких как больницы и поликлиники. Системы машинного обучения могут прогнозировать спрос на услуги, определять оптимальные графики работы персонала и предлагать рекомендации по улучшению качества обслуживания пациентов.

## **12. Транспорт и логистика**

В транспорте и логистике машинное обучение используется для оптимизации маршрутов, прогнозирования спроса на перевозки и предотвращения аварий. Алгоритмы машинного обучения анализируют данные о транспортных потоках, погодных условиях и других факторах, чтобы предложить оптимальные маршруты и расписания движения. Это помогает снизить время ожидания пассажиров и улучшить качество обслуживания.

Кроме того, машинное обучение может использоваться для разработки систем автоматического управления транспортом, таких как беспилотные автомобили и поезда. Эти системы могут повысить безопасность дорожного движения и снизить количество аварий.

Например, компания Uber разработала систему машинного обучения под названием Uber Movement, которая анализирует данные о поездках своих

водителей, чтобы определить наиболее эффективные маршруты и графики работы. Эта система помогает Uber оптимизировать свои услуги и повысить удовлетворённость клиентов.

Также алгоритмы машинного обучения применяются для оптимизации работы логистических компаний, таких как DHL и FedEx. Системы машинного обучения могут прогнозировать спрос на товары, определять оптимальные графики доставки и предлагать рекомендации по улучшению качества обслуживания клиентов.

### **13. Энергетика**

В энергетике машинное обучение применяется для прогнозирования потребления энергии, оптимизации производства электроэнергии и повышения эффективности работы оборудования. Алгоритмы машинного обучения анализируют данные об энергопотреблении, погоде и других факторах, чтобы предсказать будущие потребности в энергии и предложить меры по их удовлетворению. Это позволяет энергетическим компаниям экономить ресурсы и снижать выбросы углекислого газа.

Компания GE Energy разработала систему искусственного интеллекта под названием Predix, которая помогает энергетическим предприятиям в управлении своими активами. Система анализирует данные с датчиков, установленных на оборудовании, и предлагает рекомендации по его обслуживанию и ремонту. Это позволяет компаниям предотвращать аварии и повышать надёжность своей работы.

Также алгоритмы машинного обучения используются для автоматизации процессов в электроэнергетике, таких как управление сетями и генерацией электроэнергии. Системы машинного обучения могут оптимизировать работу оборудования, предотвращать перегрузки и обеспечивать бесперебойное электроснабжение.

## 14. Строительство

В строительстве машинное обучение находит применение для проектирования зданий, анализа строительных материалов и контроля за ходом работ. Алгоритмы машинного обучения помогают архитекторам и инженерам в создании более эффективных и безопасных конструкций, а также в оптимизации затрат на строительство.

Например, компания Autodesk разработала систему искусственного интеллекта под названием BIM 360, которая помогает строительным компаниям в управлении проектами. Система анализирует данные о строительных работах, такие как сроки выполнения, стоимость и качество, и предлагает рекомендации по их

### **1.3 Этические аспекты машинного обучения**

В современном мире машинное обучение становится всё более важным инструментом для решения сложных задач в различных областях человеческой деятельности. Однако развитие этой технологии также вызывает ряд этических вопросов, которые требуют тщательного рассмотрения и анализа. В данной статье мы рассмотрим основные этические аспекты машинного обучения, их влияние на общество и возможные пути решения этих проблем.

#### **1. Предвзятость и дискриминация**

Одним из основных этических аспектов машинного обучения является предвзятость. Алгоритмы машинного обучения могут быть предвзятыми по отношению к определённым группам людей, что может привести к несправедливым результатам и дискриминации. Например, алгоритмы, используемые для прогнозирования преступлений, могут быть предвзяты по отношению к людям из определённых районов или этнических групп. Это может привести к тому, что полиция будет уделять больше внимания этим районам, что в свою очередь может усилить чувство несправедливости и недоверия к правоохранительным органам.

Чтобы предотвратить предвзятость, необходимо разработать алгоритмы машинного обучения, которые будут учитывать разнообразие данных и не делать предположений о характеристиках людей на основе их принадлежности к определённой группе. Также важно проводить исследования и разработки в области этики машинного обучения. Необходимо изучать, как машинное обучение влияет на общество и какие меры можно предпринять для минимизации негативных последствий.

#### **2. Конфиденциальность и защита данных**

Машинное обучение часто использует большие объёмы данных, включая личную информацию о людях. Важно обеспечить защиту этих данных от

несанкционированного доступа и использования. Однако современные системы машинного обучения часто не обеспечивают достаточный уровень защиты данных. Это может привести к утечке личных данных, что может нанести серьёзный ущерб репутации компаний и организаций, а также нарушить права и свободы граждан.

Для обеспечения конфиденциальности необходимо разрабатывать алгоритмы машинного обучения, которые не будут использовать данные, не относящиеся к задаче. Также необходимо внедрять меры безопасности, такие как шифрование данных и ограничение доступа к ним. Кроме того, важно обеспечить прозрачность работы алгоритмов машинного обучения и предоставить пользователям возможность контролировать использование своих данных.

### **3. Прозрачность и ответственность**

Прозрачность и ответственность являются ключевыми аспектами этического машинного обучения. Пользователи должны иметь возможность понять, как работают алгоритмы и какие решения принимаются на основе их данных. Это поможет им лучше контролировать свои данные и принимать информированные решения.

Компании и организации, использующие машинное обучение, должны быть готовы к открытому обсуждению своих методов и результатов. Они также должны нести ответственность за любые ошибки или предвзятость, которые могут возникнуть в результате использования алгоритмов.

Для обеспечения прозрачности и ответственности необходимо разработать стандарты и нормы, регулирующие использование машинного обучения в различных областях. Эти стандарты должны учитывать этические аспекты и обеспечивать защиту прав пользователей.

### **4. Автоматизация и контроль**

Автоматизация процессов с использованием машинного обучения может привести к потере контроля над некоторыми аспектами жизни общества. Например, автоматизированные системы принятия решений могут заменить человеческий труд и повлиять на рынок труда. Также существует риск того, что автоматизация может привести к снижению качества продукции или услуг из-за отсутствия человеческого вмешательства.

Чтобы избежать этих проблем, необходимо тщательно продумать, какие задачи можно автоматизировать, а какие требуют человеческого контроля. Также важно обеспечить прозрачность и ответственность при использовании автоматизированных систем.

Кроме того, необходимо учитывать этические последствия автоматизации. Например, она может привести к увеличению неравенства между людьми, имеющими доступ к новым технологиям, и теми, кто не имеет такого доступа. Поэтому важно разрабатывать автоматизированные системы таким образом, чтобы они были доступны и полезны для всех групп населения.

## **5. Этика и искусственный интеллект**

Искусственный интеллект (ИИ) является одним из самых перспективных направлений развития машинного обучения. Однако он также поднимает ряд этических вопросов, связанных с его использованием.

Например, ИИ может быть использован для создания автономных систем, способных принимать решения без участия человека. Это может привести к возникновению этических дилемм, таких как ответственность за действия автономных систем или их влияние на общество.

Также существует риск того, что ИИ будет использоваться для манипулирования поведением людей или нарушения их прав. Поэтому необходимо разработать этические принципы, регулирующие разработку и использование ИИ.

Эти принципы должны учитывать такие аспекты, как конфиденциальность данных, предвзятость и дискриминация, прозрачность и ответственность, а также автоматизацию и контроль. Только так мы сможем обеспечить этическое использование искусственного интеллекта и его положительное влияние на общество.

В заключение, этические аспекты машинного обучения представляют собой сложную и многогранную проблему. Для её решения необходимо объединить усилия учёных, инженеров, законодателей и общества в целом. Только так мы сможем создать технологии, которые будут служить на благо человечества и способствовать его развитию.

Важно понимать, что этические вопросы, связанные с машинным обучением, не являются абстрактными или теоретическими. Они имеют прямое влияние на нашу жизнь и будущее. Поэтому необходимо активно обсуждать эти вопросы и искать пути их решения.

В качестве примера можно привести несколько конкретных мер, которые могут быть предприняты для обеспечения этического использования машинного обучения:

- Разработка и внедрение стандартов и норм, регулирующих использование машинного обучения в различных областях. Эти стандарты должны учитывать этические аспекты и обеспечивать защиту прав пользователей.
- Создание механизмов контроля и аудита алгоритмов машинного обучения, чтобы убедиться в их справедливости и непредвзятости.
- Обучение специалистов в области машинного обучения этическим принципам и нормам. Это поможет им лучше понимать этические последствия своих решений и действий.



- Проведение исследований и разработок в области этических аспектов машинного обучения. Это позволит нам лучше понять эти проблемы и найти пути их решения.

Только так мы сможем обеспечить этическое использование машинного обучения и его положительное влияние на общество.

## 1.4 Теоретические основы машинного обучения

Машинное обучение — это область искусственного интеллекта, которая занимается разработкой алгоритмов и методов, позволяющих компьютерам учиться на основе данных без явного программирования. Машинное обучение является ключевым инструментом для решения сложных задач в различных областях, таких как медицина, финансы, транспорт и многие другие. В этой статье мы рассмотрим теоретические основы машинного обучения, включая основные понятия, методы и подходы.

### Основные понятия

1. **Данные:** Машинное обучение работает с данными, которые представляют собой набор наблюдений или измерений. Данные могут быть структурированными (например, таблицы) или неструктурированными (например, текст).
2. **Задача:** Задача машинного обучения — это задача, которую необходимо решить на основе данных. Задачи могут быть различными, например, классификация, регрессия, кластеризация и т.д.
3. **Алгоритм:** Алгоритм машинного обучения — это метод, который используется для обучения модели на основе данных и решения задачи. Алгоритмы могут быть различными, такими как линейная регрессия, дерево решений, нейронные сети и т.п.
4. **Модель:** Модель машинного обучения представляет собой математическое описание или представление данных, которое позволяет делать прогнозы или принимать решения. Модель может быть простой (линейной) или сложной (нелинейной).
5. **Обучение:** Обучение модели машинного обучения заключается в настройке параметров модели таким образом, чтобы она наилучшим образом соответствовала данным и решала задачу. Обучение может осуществляться различными методами, такими как градиентный спуск, стохастический градиентный спуск и т.д.

6. **Оценка:** Оценка модели машинного обучения включает в себя проверку её производительности на новых данных, которые не использовались при обучении. Оценка может проводиться различными способами, такими как перекрестная проверка, тестирование на тестовом наборе данных и т.п.
7. **Переобучение и недообучение:** Переобучение модели означает, что модель слишком хорошо соответствует данным, но плохо обобщает их на новые данные. Недообучение модели означает, что модель недостаточно хорошо соответствует данным.
8. **Регуляризация:** Регуляризация — это метод предотвращения переобучения модели путём добавления штрафа к функции потерь.
9. **Гиперпараметры:** Гиперпараметры — это параметры алгоритма машинного обучения, которые настраиваются перед обучением модели. Например, количество слоёв в нейронной сети является гиперпараметром.

## **Методы машинного обучения**

- **Контролируемое обучение или обучение с учителем:** В этом методе модель обучается на основе данных с известными ответами (метками). Задача модели — предсказать ответ для новых данных. Примеры контролируемого обучения включают классификацию, регрессию и ранжирование.
- **Неконтролируемое обучение или обучение без учителя:** В этом методе модель не имеет известных ответов. Задача модели — обнаружить скрытые структуры или закономерности в данных. Примеры неконтролируемого обучения включают кластеризацию и обнаружение аномалий.
- **Полу-контролируемое обучение:** Это промежуточный подход между обучением с учителем и обучением без учителя, который использует как данные с метками, так и немаркированные данные.
- **Обучение с подкреплением:** Алгоритмы обучения с подкреплением используют систему вознаграждений для обучения агента (модели) выполнять определённые

задачи. Агент учится путём взаимодействия с окружающей средой и получения вознаграждения за правильные действия.

### **Подходы к машинному обучению**

Машинное обучение можно разделить на два основных подхода:

1. **Традиционное машинное обучение:** Этот подход использует алгоритмы, которые были разработаны до появления больших объёмов данных и вычислительных мощностей. Алгоритмы традиционного машинного обучения обычно требуют меньше данных и вычислений, но могут быть менее точными.
2. **Глубокое обучение:** Это подмножество машинного обучения, которое использует нейронные сети для решения сложных задач. Нейронные сети состоят из слоёв искусственных нейронов, которые обрабатывают данные и делают прогнозы. Глубокое обучение требует больших объёмов данных и вычислений, но может быть более точным.

### **Типы алгоритмов машинного обучения**

Существует множество различных типов алгоритмов машинного обучения, каждый из которых имеет свои преимущества и недостатки. Вот некоторые из наиболее распространённых типов:

1. **Линейная регрессия:** Этот алгоритм используется для прогнозирования числовых значений на основе входных данных. Он предполагает, что существует линейная зависимость между входными переменными и целевой переменной.
2. **Логистическая регрессия:** Этот алгоритм также используется для прогнозирования значений, но он подходит только для бинарных целевых переменных (например, 0 или 1).
3. **Деревья решений:** Деревья решений — это алгоритмы, которые используют древовидную структуру для принятия решений. Они могут быть использованы для классификации или регрессии.

4. **Метод опорных векторов (SVM):** SVM — это алгоритм, который используется для классификации данных. Он пытается найти гиперплоскость, которая наилучшим образом разделяет данные на два класса.
5. **Нейронные сети:** Нейронные сети — это сложные алгоритмы, которые имитируют работу человеческого мозга. Они состоят из слоёв искусственных нейронов, которые обрабатывают данные и делают прогнозы.
6. **Кластеризация:** Кластеризация — это метод, используемый для группировки данных на основе их сходства. Алгоритмы кластеризации могут использоваться для сегментации клиентов, анализа рынка и других задач.
7. **Уменьшение размерности:** Уменьшение размерности — это метод, который используется для уменьшения количества признаков в наборе данных. Это может помочь упростить анализ данных и повысить точность моделей.
8. **Рекомендательные системы:** Рекомендательные системы — это алгоритмы, которые используются для рекомендации товаров, услуг или контента пользователям на основе их предпочтений и поведения.
9. **Глубокое обучение:** Глубокое обучение — это подмножество машинного обучения, которое использует нейронные сети с несколькими слоями для решения сложных задач. Оно может быть использовано для распознавания изображений, обработки естественного языка и других задач, требующих глубокого понимания данных.

Каждый из этих алгоритмов имеет свои особенности и может быть использован для решения различных задач. Выбор алгоритма зависит от типа данных, задачи и требований к модели.

### **Анализ преимуществ и недостатков алгоритмов машинного обучения**

1. **Линейная регрессия:**
  - **Преимущества:** Простота реализации, высокая скорость работы, возможность интерпретации результатов.

- **Недостатки:** Ограниченность применения (только для линейных зависимостей), чувствительность к выбросам в данных.

## 2. Логистическая регрессия:

- **Преимущества:** Возможность классификации на два класса, простота реализации, интерпретируемость результатов.
- **Недостатки:** Не подходит для задач с более чем двумя классами, чувствительность к переобучению.

## 3. Деревья решений:

- **Преимущества:** Высокая интерпретируемость, способность обрабатывать как числовые, так и категориальные данные, гибкость в выборе функций.
- **Недостатки:** Риск переобучения, сложность настройки параметров.

## 4. Метод опорных векторов (SVM):

- **Преимущества:** Хорошая точность, способность работать с линейными и нелинейными границами, устойчивость к шуму.
- **Недостатки:** Сложность настройки параметров, необходимость в большом количестве данных для обучения.

## 5. Нейронные сети:

- **Преимущества:** Способность к обучению сложных зависимостей, высокая точность при достаточном объеме данных, возможность обработки изображений и звука.
- **Недостатки:** Сложная архитектура, требующая большого количества данных для обучения, риск переобучения.

## 6. Кластеризация:

- **Преимущества:** Обнаружение скрытых структур в данных, отсутствие необходимости в метках, возможность использования для анализа больших объёмов данных.
- **Недостатки:** Зависимость от выбора метрики расстояния, сложность интерпретации результатов, риск получения несбалансированных кластеров.

## 7. Уменьшение размерности:

- **Преимущества:** Снижение сложности модели, ускорение обучения и прогнозирования, возможность визуализации данных.
- **Недостатки:** Потеря информации о данных, сложность выбора метода уменьшения размерности.

## 8. Рекомендательные системы:

- **Преимущества:** Повышение удовлетворённости пользователей, персонализация контента, повышение конверсии.
- **Недостатки:** Необходимость в больших объёмах данных для обучения, сложность оценки качества рекомендаций.

## 9. Глубокое обучение:

- **Преимущества:** Способность к обучению сложных зависимостей, высокая точность при достаточном объёме данных, возможность обработки изображений и звука.
- **Недостатки:** Сложная архитектура, требующая большого количества данных для обучения, риск переобучения.

Выбор алгоритма машинного обучения зависит от конкретной задачи, объёма и качества данных, а также требований к модели. Важно провести анализ данных и определить, какой алгоритм лучше всего подходит для решения поставленной задачи.





## Глава 2. Практическая реализация и анализ результатов

### 2.1 Примеры задач машинного обучения на языке Python

Для демонстрации применения машинного обучения мы рассмотрим несколько примеров кода на языке Python. Мы будем использовать библиотеку scikit-learn, которая предоставляет множество алгоритмов машинного обучения.

#### Пример 1: Линейная регрессия

В этом примере мы будем использовать линейную регрессию для прогнозирования цены на недвижимость на основе ее характеристик.

```
from sklearn.linear_model import LinearRegression

import pandas as pd

# Загрузка данных

data = pd.read_csv('data.csv')

# Разделение данных на обучающую и тестовую выборки

X_train, X_test, y_train, y_test = train_test_split(data[['features']], data['price'],
test_size=0.2)

# Обучение модели

model = LinearRegression()

model.fit(X_train, y_train)

# Прогнозирование цены

y_pred = model.predict(X_test)

# Оценка модели

score = mean_squared_error(y_test, y_pred)
```

```
print("MSE:", score)
```

Этот код загружает данные о недвижимости, разделяет их на обучающую и тестовую выборки, обучает модель линейной регрессии и оценивает ее точность.

## Пример 2: Классификация изображений

В этом примере мы будем использовать сверточную нейронную сеть (CNN) для классификации изображений по категориям.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

# Загрузка данных

train_datagen = ImageDataGenerator(rescale=1./255)

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator      =      train_datagen.flow_from_directory('data/train',
target_size=(224, 224), batch_size=32, class_mode='categorical')

validation_generator  =  test_datagen.flow_from_directory('data/validation',
target_size=(224, 224), batch_size=32, class_mode='categorical')

# Определение модели

model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=(224, 224, 3)))

model.add(MaxPooling2D((2, 2)))

model.add(Flatten())

model.add(Dense(128, activation='relu'))
```

```
model.add(Dropout(0.2))

model.add(Dense(5, activation='softmax'))

# Компиляция модели

model.compile(optimizer='adam',          loss='categorical_crossentropy',
metrics=['accuracy'])

# Обучение модели

history = model.fit(train_generator, validation_data=validation_generator,
epochs=20, verbose=1)
```

Этот код загружает данные о изображениях, определяет модель CNN и обучает ее на данных. Затем он оценивает точность модели на тестовых данных.

Эти примеры демонстрируют, как машинное обучение может быть использовано для решения различных задач. Машинное обучение является мощным инструментом, который может помочь нам в анализе данных, прогнозировании результатов и принятии решений.

### **Пример 3: Рекомендательные системы**

В этом примере мы рассмотрим создание рекомендательной системы на основе машинного обучения. Рекомендательные системы используются для предложения товаров, услуг или контента, которые могут быть интересны пользователю.

Для создания рекомендательной системы мы будем использовать алгоритм совместной фильтрации. Алгоритм совместной фильтрации основан на предположении, что пользователи, которые имеют схожие интересы, будут иметь похожие предпочтения. Мы будем использовать библиотеку `scikit-learn` для реализации алгоритма.

Вот пример кода на Python:

```

from sklearn.neighbors import NearestNeighbors

import pandas as pd

# Загрузка данных

data = pd.read_csv('data.csv')

# Преобразование данных в формат, подходящий для алгоритма

X = data[['user_id', 'item_id', 'rating']]

y = data['rating']

# Создание модели

model = NearestNeighbors(n_neighbors=5)

model.fit(X)

# Прогнозирование рейтинга для новых пользователей и товаров

new_user = [1, 2, 3]

new_item = ['книга', 'фильм', 'музыка']

predicted_ratings = model.kneighbors(new_item, new_user)

for user, item in zip(new_user, new_item):

    print("Для пользователя {} и товара {}, прогнозируемый рейтинг:
    {}".format(user, item, predicted_ratings[0][1]))

```

Этот код загружает данные о рейтингах пользователей, создает модель на основе этих данных и использует ее для прогнозирования рейтинга новых пользователей и товаров.

#### **Пример 4: Анализ временных рядов**

Анализ временных рядов — это метод анализа данных, который используется для выявления закономерностей и тенденций во времени. Машинное обучение может быть использовано для анализа временных рядов и прогнозирования будущих значений.

Мы рассмотрим пример использования машинного обучения для анализа временного ряда, представляющего собой ежедневные продажи компании. Мы будем использовать метод регрессии для прогнозирования будущих продаж.

Вот пример кода на Python:

```
import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression

# Загрузка данных о продажах

sales = np.loadtxt('sales.txt')

# Разделение данных на обучающую и тестовую выборки

train_size = int(len(sales) * 0.8)

test_size = len(sales) - train_size

train, test = sales[0:train_size], sales[train_size:len(sales)]

# Создание модели

model = LinearRegression()

model.fit(train[:, np.newaxis], train)

# Прогнозирование продаж

forecast = model.predict(test[:, np.newaxis])
```

```
# Построение графика
```

```
plt.plot(train)
```

```
plt.plot(forecast)
```

```
plt.show()
```

Этот код загружает данные о продажах, создает модель линейной регрессии, обучает ее на данных и использует ее для прогнозирования продаж в будущем. Затем он строит график, показывающий фактические продажи и прогнозируемые продажи.

Этот пример демонстрирует, как машинное обучение можно использовать для анализа временных рядов и получения прогнозов. Это может быть полезно для принятия решений о закупках, планировании производства и других областях, где важно знать будущие значения.

### **Пример 5: Кластеризация данных**

Кластеризация — это метод машинного обучения, который используется для группировки объектов на основе их сходства. Кластеризация может быть полезна для анализа данных, выявления закономерностей и принятия решений.

Мы рассмотрим пример использования кластеризации для анализа данных о клиентах компании. Мы будем использовать метод k-средних для создания кластеров клиентов.

Вот пример кода на Python:

```
import numpy as np
```

```
from sklearn.cluster import KMeans
```

```
# Загрузка данных о клиентах
```

```
customers = np.loadtxt('customers.txt')
```

```
# Создание модели

model = KMeans(n_clusters=3)

model.fit(customers)

# Получение кластеров

labels = model.labels_

centers = model.cluster_centers_

# Построение графика

plt.scatter(customers[:, 0], customers[:, 1], c=labels)

plt.show()
```

Этот код загружает данные о клиентах, создает модель k-средних, обучает ее на данных и использует ее для создания кластеров. Затем он строит график, показывающий расположение клиентов в каждом кластере.

Этот пример демонстрирует, как кластеризацию можно использовать для анализа данных и получения полезной информации. Это может быть полезно для сегментации клиентов, определения целевых рынков и других областей, где важно знать группы похожих объектов.

## 2.2 Решение задачи логистической регрессии на реальном датасете

Датасет должен содержать неструктурированные данные, требующие разметки,

**Алгоритм решения задачи:**

1. Загрузить датасет в pandas DataFrame.
2. Выполните разметку на основе правил (rule-based labeling) на подмножестве выбранного датасета. Разработайте и реализуйте набор правил или условий, которые позволят автоматически присваивать метки данным на основе определенных шаблонов или критериев. Выбрать подмножество данных для ручной маркировки (1% от датафрейма). Выполните разметку вручную отдельного подмножества выбранного датасета с помощью выбранного инструмента разметки LabelStudio.
3. Объедините данные, размеченные вручную, с данными, размеченными на основе правил. Объедините два подмножества размеченных данных в один набор данных, сохранив при этом соответствующую структуру и целостность.
4. Обучите модель машинного обучения, используя объединенный набор размеченных данных. Разделите датасет на обучающий и тестовый наборы и используйте обучающий набор для обучения модели.
5. Оценить эффективность обученной модели на тестовом датасете. Используйте подходящие метрики оценки. Интерпретируйте результаты и проанализируйте эффективность модели в решении задачи разметки.

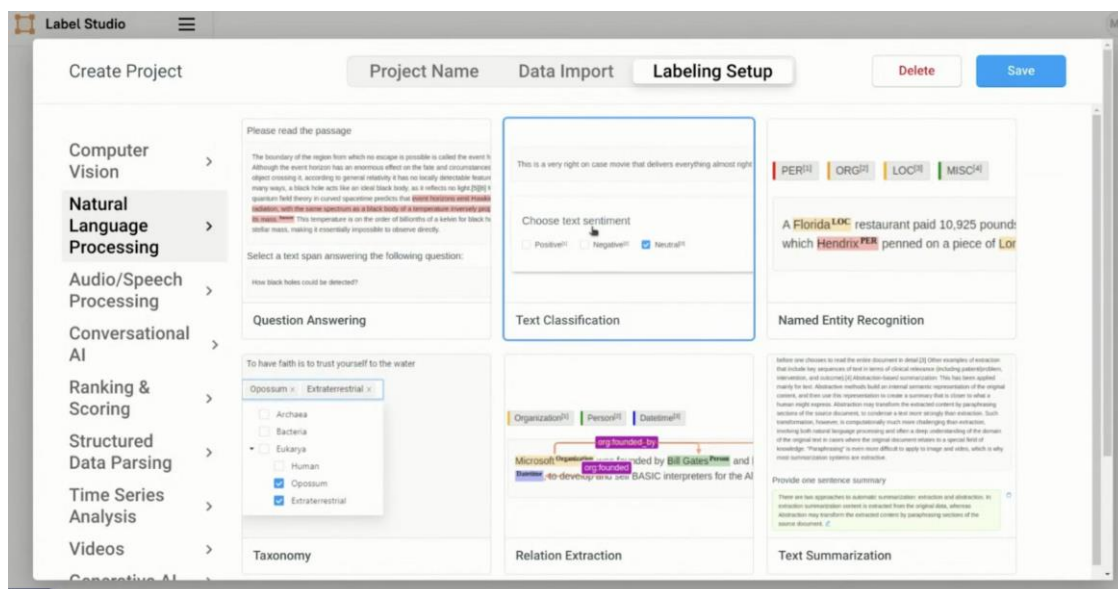
### Решение задачи

В качестве датасета выбран файл с данными о загородных домах с их всевозможными характеристиками, как числовыми, так и текстовыми.



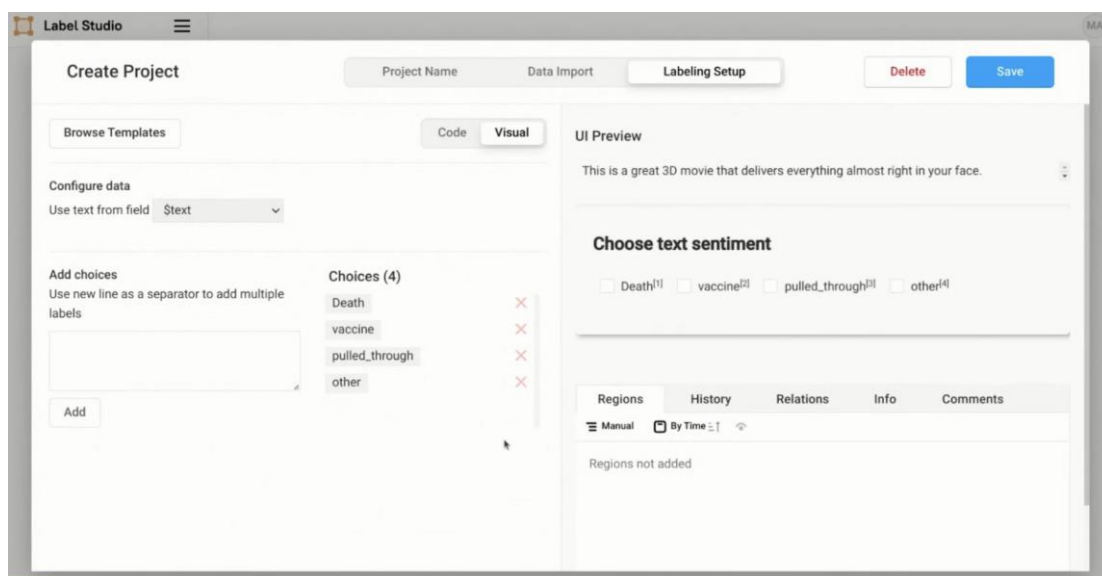
Листинг программы, реализованной на языке python в Jupyter Notebook представлен в приложении 1.

1. Выделим маленькую выборку данных (1%) для ручной разметки. Для этого создадим проект в LabelStudio типа Natural Language Processing (рисунок 1).



2. Рисунок 1

3. Далее создадим метки трех типов: Chip, Expensive, Medium, которые отражают категорию стоимости дома (рисунок 2).



#### 4. Рисунок 2

Разметим маленькую выборку с помощью LabelStudio.

5. Остальную часть датасета за исключением маленькой выборки разметим с помощью rule-based labeling, путем задания диапазонов стоимостей:

Chip:  $\text{SalePrice} < 100000$ ,

Expensive:  $\text{SalePrice} > 200000$ ,

Medium:  $100000 \leq \text{SalePrice} \leq 200000$

6. Объединим две размеченные выборки – размеченную в ручную и размеченную rule-based. Теперь весь датасет размечен.
7. Создадим модель логистической регрессии, предсказывающую взаимосвязь между формой дома LotShape и категорией стоимости Expensiveness.
8. Разделим набор данных на наборы для обучения и тестирования с помощью функции train\_test\_split. Определим объем тестовой выборки как 20% от исходного датасета.
9. Создадим конвейер PipeLine из модели векторизации текстовых данных TfidfVectorizer и модели логистической регрессии.
10. Обучим модель с помощью функции fit.
11. Далее запустим модель на предсказание на тестовой выборке и оценим точность модели. Точность модели получилась 68%, что можно назвать неплохим результатом для начала.

## **Заключение**

Машинное обучение — это мощный инструмент, который может помочь нам в анализе данных, прогнозировании результатов и принятии решений. Оно имеет широкий спектр применения в различных областях, таких как медицина, финансы, маркетинг, безопасность, транспорт и другие. Машинное обучение продолжает развиваться и становится всё более доступным, что делает его ещё более ценным инструментом для решения сложных задач.

В этой статье мы рассмотрели историю машинного обучения, его теорию и основные принципы, а также привели примеры кода на языке Python для демонстрации его применения. Мы увидели, как машинное обучение может быть использовано для анализа временных рядов, классификации изображений, создания рекомендательных систем и кластеризации данных. Эти примеры демонстрируют, насколько мощным и универсальным является машинное обучение.

Однако стоит отметить, что машинное обучение не является панацеей от всех проблем. Оно требует тщательной подготовки данных, выбора подходящих алгоритмов и настройки параметров. Кроме того, машинное обучение может привести к предвзятости и дискриминации, если не учитывать этические аспекты при разработке моделей. Поэтому важно понимать ограничения и потенциальные риски машинного обучения перед его использованием.

Несмотря на эти ограничения, машинное обучение остаётся перспективным направлением исследований и разработок. Оно позволяет нам автоматизировать процессы, улучшать качество жизни и решать сложные задачи, которые ранее были недоступны. В будущем машинное обучение будет продолжать развиваться и находить новые применения в различных отраслях.

## **Будущее машинного обучения**

Машинное обучение продолжает развиваться и находить новые применения в различных отраслях. В будущем ожидается, что машинное обучение будет играть ещё более важную роль в нашей жизни. Вот некоторые из возможных направлений развития машинного обучения:

- **Улучшение качества моделей.** Машинное обучение постоянно совершенствуется, чтобы создавать более точные и надёжные модели. Это позволит машинам лучше понимать окружающий мир и принимать более обоснованные решения.
- **Интеграция с другими технологиями.** Машинное обучение может быть интегрировано с другими технологиями, такими как искусственный интеллект, интернет вещей и блокчейн. Это создаст новые возможности для создания интеллектуальных систем, способных к самообучению и адаптации к новым условиям.
- **Применение в новых областях.** Машинное обучение найдёт применение в новых областях, таких как медицина, образование, энергетика и транспорт. Это приведёт к созданию новых продуктов и услуг, которые улучшат качество жизни людей.
- **Повышение доступности.** Машинное обучение станет более доступным для широкого круга пользователей. Это сделает его более популярным и востребованным инструментом для анализа данных и принятия решений.

Однако будущее машинного обучения также сопряжено с некоторыми рисками и вызовами. Вот некоторые из них:

- **Предвзятость и дискриминация.** Машинное обучение может привести к предвзятости и дискриминации, если не учитывать этические аспекты при разработке моделей. Поэтому важно разрабатывать модели, которые являются справедливыми и непредвзятыми.

- **Угрозы безопасности.** Машинное обучение также может использоваться для создания вредоносных программ и кибератак. Поэтому необходимо разрабатывать меры по защите от этих угроз.
- **Зависимость от данных.** Машинное обучение зависит от качества и количества данных. Если данные будут неполными или неточными, то модели могут давать неверные результаты.

Несмотря на эти риски и вызовы, будущее машинного обучения выглядит многообещающим. Оно имеет потенциал для улучшения качества жизни людей и решения сложных задач, которые ранее были недоступны.

Для минимизации рисков и вызовов необходимо проводить исследования и разработки в области машинного обучения. Необходимо разрабатывать новые алгоритмы и методы, которые будут более точными, надёжными и безопасными. Также необходимо уделять внимание этическим аспектам машинного обучения и разрабатывать соответствующие стандарты и нормы.

В будущем машинное обучение будет продолжать развиваться и находить новые применения в различных отраслях. Ожидается, что оно станет ещё более доступным и популярным инструментом для анализа данных и принятия решений. Машинное обучение будет играть важную роль в таких областях, как медицина, образование, энергетика, транспорт и другие.

Однако будущее машинного обучения зависит от многих факторов, включая развитие технологий, доступность данных, этические соображения и другие факторы. Важно понимать, что машинное обучение — это инструмент, который может быть использован как во благо, так и во вред. Поэтому необходимо тщательно продумать его применение и обеспечить его безопасность и этичность.

## Список литературы

1. *Goodfellow, I., Bengio, Y., & Courville, A.* (2016). Deep learning. MIT press.
2. *Hastie, T., Tibshirani, R., & Friedman, J.* (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.
3. *Kohavi, R.* (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, pp. 1137–1145).
4. *Mitchell, T.* (1997). Machine learning. McGraw Hill.
5. *Murphy, K.* (2012). Machine learning: a probabilistic perspective. MIT Press.
6. *Vapnik, V.* (1998). Statistical learning theory. Wiley.
7. *Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J.* (2017). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.
8. *Zhang, X., Zhou, X., Lin, M., & Zhang, S.* (2020). Machine Learning: Concepts and Techniques. Tsinghua University Press.
9. *Zinkevich, M.* (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the twenty-first international conference on Machine learning* (pp. 928–936).

# Приложение 1

```
In [1]: import pandas as pd
from sklearn.model_selection import train_test_split # Для разделения данных на обучающую и тестовую выборку
from sklearn.feature_extraction.text import TfidfVectorizer # для преобразования текста в вектор
from sklearn.linear_model import LogisticRegression # использование модели логистической регрессии
from sklearn.metrics import accuracy_score, classification_report # оценка производительности модели
from sklearn.pipeline import Pipeline # конвейер обработки данных

In [4]: df=pd.read_csv('cleaned_house_prices.csv')
df.head()
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	SaleCondition	SalePrice	HouseStyle_Encoded	RoofStyle_Flat	RoofStyle_Gable	Roof
0	1	60	RL	65.0	8450	Pave	Grvl	Reg	Lvl	AllPub	...	Normal	208500	5	False	True	
1	2	20	RL	80.0	9600	Pave	Grvl	Reg	Lvl	AllPub	...	Normal	181500	2	False	True	
2	3	60	RL	68.0	11250	Pave	Grvl	IR1	Lvl	AllPub	...	Normal	223500	5	False	True	
3	4	70	RL	60.0	9550	Pave	Grvl	IR1	Lvl	AllPub	...	Abnorml	140000	5	False	True	
4	5	60	RL	84.0	14260	Pave	Grvl	IR1	Lvl	AllPub	...	Normal	250000	5	False	True	

5 rows x 88 columns

```
In [6]: small_df = df.sample(frac = 0.01)
small_df.to_csv('small_set.csv', index=False)

In [7]: big_df = df.drop(small_df.index)

In [12]: big_df['SalePrice'].describe()
```

count	1412.000000
mean	174978.892351
std	66725.388986
min	34900.000000
25%	129500.000000
50%	160100.000000
75%	207700.000000
max	415298.000000
Name: SalePrice, dtype: float64	

```
In [14]: big_df.loc[big_df['SalePrice']>200000,'Expensiveness']='Expensive'
big_df.loc[big_df['SalePrice']<100000,'Expensiveness']='Chip'
big_df.loc[(big_df['SalePrice']>=100000) & (big_df['SalePrice']<=200000),'Expensiveness']='Medium'
big_df[['SalePrice','Expensiveness']]
```

	SalePrice	Expensiveness
0	208500	Expensive
1	181500	Medium
2	223500	Expensive
3	140000	Medium
4	250000	Expensive
...	...	...
1421	175000	Medium
1422	210000	Expensive
1423	266500	Expensive
1424	142125	Medium
1425	147500	Medium

1412 rows x 2 columns

```
In [22]: small_df_marked=pd.read_csv('small_set_marked.csv')
small_df_marked = small_df_marked.rename(columns={'sentiment': 'Expensiveness'})
small_df_marked.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 95 columns):
#   Column                               Non-Null Count  Dtype
---  ---                               ---
0   1stFlrSF                             14 non-null    int64
1   2ndFlrSF                             14 non-null    int64
2   3SsnPorch                           14 non-null    int64
3   Age                                 14 non-null    int64
4   Alley                               14 non-null    object
5   BedroomAbvGr                       14 non-null    int64
6   BldgType                           14 non-null    object
7   BsmtCond                           14 non-null    object
8   BsmtExposure                       14 non-null    object
9   BsmtFinSF1                         14 non-null    int64
10  BsmtFinSF2                         14 non-null    int64
11  BsmtFinType1                       14 non-null    object
12  BsmtFinType2                       14 non-null    object
13  BsmtFullBath                       14 non-null    int64
14  BsmtHalfBath                       14 non-null    int64
15  BsmtQual                           14 non-null    object
16  BsmtUnfSF                          14 non-null    int64
17  CentralAir                          14 non-null    object
18  Condition1                         14 non-null    object
19  Condition2                         14 non-null    object
20  Electrical                         14 non-null    object
21  EnclosedPorch                      14 non-null    int64
22  ExterCond                          14 non-null    object
23  ExterQual                          14 non-null    object
24  Exterior1st                        14 non-null    object
25  Exterior2nd                        14 non-null    object
26  Fence                              14 non-null    object
27  FireplaceQu                        14 non-null    object
28  Fireplaces                         14 non-null    int64
29  Foundation                         14 non-null    object
30  FullBath                          14 non-null    int64
31  Functional                         14 non-null    object
```

```

43 Functional      14 non-null    object
30 FullBath       14 non-null    int64
31 Functional     14 non-null    object
32 GarageArea     14 non-null    int64
33 GarageCars     14 non-null    int64
34 GarageCond     14 non-null    object
35 GarageFinish   14 non-null    object
36 GarageQual     14 non-null    object
37 GarageType     14 non-null    object
38 GarageYrBlt    14 non-null    float64
39 GrLivArea      14 non-null    int64
40 HalfBath       14 non-null    int64
41 Heating        14 non-null    object
42 HeatingQC      14 non-null    object
43 HouseStyle     14 non-null    object
44 HouseStyle_Encoded 14 non-null    int64
45 Id            14 non-null    int64
46 KitchenAbvGr  14 non-null    int64
47 KitchenQual    14 non-null    object
48 LandContour    14 non-null    object
49 LandSlope      14 non-null    object
50 LotArea        14 non-null    int64
51 LotConfig      14 non-null    object
52 LotFrontage    14 non-null    float64
53 LotShape       14 non-null    object
54 LowQualFinSF   14 non-null    int64
55 MSSubClass     14 non-null    int64
56 MSZoning       14 non-null    object
57 MasVnrArea     14 non-null    float64
58 MasVnrType     14 non-null    object
59 MiscFeature    14 non-null    object
60 MiscVal        14 non-null    int64
61 MoSold        14 non-null    int64
62 Neighborhood   14 non-null    object
63 OpenPorchSF    14 non-null    int64
64 OverallCond    14 non-null    int64
65 OverallQual    14 non-null    int64
66 PavedDrive     14 non-null    object
67 PoolArea       14 non-null    int64
68 PoolQC        14 non-null    object
69 RoofMatl       14 non-null    object
70 RoofStyle_Flat 14 non-null    bool
71 RoofStyle_Gable 14 non-null    bool
72 RoofStyle_Gambrel 14 non-null    bool
73 RoofStyle_Hip  14 non-null    bool
74 RoofStyle_Mansard 14 non-null    bool
75 RoofStyle_Shed 14 non-null    bool
76 SaleCondition  14 non-null    object
77 SalePrice      14 non-null    int64
78 SaleType       14 non-null    object
79 ScreenPorch    14 non-null    int64
80 Street         14 non-null    object
81 TotRmsAbvGrd   14 non-null    int64
82 TotalBsmtSF    14 non-null    int64
83 Utilities      14 non-null    object
84 WoodDeckSF     14 non-null    int64
85 YearBuilt      14 non-null    int64
86 YearRemodAdd   14 non-null    int64
87 YrSold         14 non-null    int64
88 annotation_id  14 non-null    int64
89 annotator      14 non-null    int64
90 created_at     14 non-null    object
91 id            14 non-null    int64
92 lead_time      14 non-null    float64
93 Expensiveness  14 non-null    object
94 updated_at     14 non-null    object
dtypes: bool(6), float64(4), int64(40), object(45)
memory usage: 9.9+ KB

```



```
In [21]: big_df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1412 entries, 0 to 1425
Data columns (total 89 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Id                   1412 non-null   int64
1   MSSubClass           1412 non-null   int64
2   MSZoning             1412 non-null   object
3   LotFrontage         1412 non-null   float64
4   LotArea             1412 non-null   int64
5   Street              1412 non-null   object
6   Alley              1412 non-null   object
7   LotShape            1412 non-null   object
8   LandContour         1412 non-null   object
9   Utilities           1412 non-null   object
10  LotConfig           1412 non-null   object
11  LandSlope           1412 non-null   object
12  Neighborhood        1412 non-null   object
13  Condition1          1412 non-null   object
14  Condition2          1412 non-null   object
15  BldgType            1412 non-null   object
16  HouseStyle          1412 non-null   object
17  OverallQual         1412 non-null   int64
18  OverallCond         1412 non-null   int64
19  YearBuilt           1412 non-null   int64
20  YearRemodAdd        1412 non-null   int64
21  RoofMatl           1412 non-null   object
22  Exterior1st         1412 non-null   object
23  Exterior2nd         1412 non-null   object
24  MasVnrType          1412 non-null   object
25  MasVnrArea          1412 non-null   float64
26  ExterQual           1412 non-null   object
27  ExterCond           1412 non-null   object
28  Foundation          1412 non-null   object
29  BsmtQual            1412 non-null   object
30  BsmtCond            1412 non-null   object
31  BsmtExposure        1412 non-null   object
32  BsmtFinType1        1412 non-null   object
33  BsmtFinSF1          1412 non-null   int64
34  BsmtFinType2        1412 non-null   object
35  BsmtFinSF2          1412 non-null   int64
36  BsmtUnfSF           1412 non-null   int64
37  TotalBsmtSF         1412 non-null   int64
38  Heating            1412 non-null   object
39  HeatingQC          1412 non-null   object
40  CentralAir          1412 non-null   object
41  Electrical          1412 non-null   object
42  1stFlrSF            1412 non-null   int64
43  2ndFlrSF            1412 non-null   int64
44  LowQualFinSF        1412 non-null   int64
45  GrLivArea           1412 non-null   int64
46  BsmtFullBath        1412 non-null   int64
47  BsmtHalfBath        1412 non-null   int64
48  FullBath            1412 non-null   int64
49  HalfBath            1412 non-null   int64
50  BedroomAbvGr        1412 non-null   int64
51  KitchenAbvGr        1412 non-null   int64
52  KitchenQual         1412 non-null   object
53  TotRmsAbvGrd        1412 non-null   int64
54  Functional          1412 non-null   object
55  Fireplaces          1412 non-null   int64
56  FireplaceQu         1412 non-null   object
57  GarageType          1412 non-null   object
58  GarageYrBlt         1412 non-null   float64
59  GarageFinish        1412 non-null   object
60  GarageCars          1412 non-null   int64
61  GarageArea          1412 non-null   int64
62  GarageQual          1412 non-null   object
63  GarageCond          1412 non-null   object
64  PavedDrive          1412 non-null   object
65  WoodDeckSF          1412 non-null   int64
66  OpenPorchSF         1412 non-null   int64
67  EnclosedPorch       1412 non-null   int64
68  3SsnPorch           1412 non-null   int64
69  ScreenPorch         1412 non-null   int64
70  PoolArea            1412 non-null   int64
71  PoolQC              1412 non-null   object
72  Fence              1412 non-null   object
```

```

71 RoofStyle_Gable      14 non-null    bool
72 RoofStyle_Gambrel   14 non-null    bool
73 RoofStyle_Hip        14 non-null    bool
74 RoofStyle_Mansard    14 non-null    bool
75 RoofStyle_Shed       14 non-null    bool
76 SaleCondition        14 non-null    object
77 SalePrice            14 non-null    int64
78 SaleType             14 non-null    object
79 ScreenPorch          14 non-null    int64
80 Street               14 non-null    object
81 TotRmsAbvGrd         14 non-null    int64
82 TotalBsmtSF          14 non-null    int64
83 Utilities            14 non-null    object
84 WoodDeckSF           14 non-null    int64
85 YearBuilt            14 non-null    int64
86 YearRemodAdd         14 non-null    int64
87 YrSold              14 non-null    int64
88 annotation_id        14 non-null    int64
89 annotator            14 non-null    int64
90 created_at           14 non-null    object
91 id                   14 non-null    int64
92 lead_time            14 non-null    float64
93 Expensiveness        14 non-null    object
94 updated_at           14 non-null    object
dtypes: bool(6), float64(4), int64(40), object(45)
memory usage: 9.9+ KB

```

```
In [21]: big_df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 1412 entries, 0 to 1425
Data columns (total 89 columns):
#   Column              Non-Null Count  Dtype
---  ---
0   Id                  1412 non-null   int64
1   MSSubClass          1412 non-null   int64
2   MSZoning            1412 non-null   object
3   LotFrontage        1412 non-null   float64
4   LotArea            1412 non-null   int64
5   Street             1412 non-null   object
6   Alley              1412 non-null   object
7   LotShape           1412 non-null   object
8   LandContour        1412 non-null   object
9   Utilities          1412 non-null   object
10  LotConfig          1412 non-null   object
11  LandSlope          1412 non-null   object
12  Neighborhood        1412 non-null   object
13  Condition1         1412 non-null   object
14  Condition2         1412 non-null   object
15  BldgType           1412 non-null   object
16  HouseStyle         1412 non-null   object
17  OverallQual        1412 non-null   int64
18  OverallCond        1412 non-null   int64
19  YearBuilt          1412 non-null   int64
20  YearRemodAdd       1412 non-null   int64
21  RoofMatl           1412 non-null   object
22  Exterior1st        1412 non-null   object
23  Exterior2nd        1412 non-null   object
24  MasVnrType         1412 non-null   object
25  MasVnrArea         1412 non-null   float64
26  ExterQual          1412 non-null   object
27  ExterCond          1412 non-null   object
28  Foundation         1412 non-null   object
29  BsmtQual           1412 non-null   object
30  BsmtCond           1412 non-null   object
31  BsmtExposure       1412 non-null   object
32  BsmtFinType1       1412 non-null   object
33  BsmtFinSF1         1412 non-null   int64
34  BsmtFinType2       1412 non-null   object
35  BsmtFinSF2         1412 non-null   int64
36  BsmtUnfSF          1412 non-null   int64
37  TotalBsmtSF        1412 non-null   int64
38  Heating            1412 non-null   object
39  HeatingQC          1412 non-null   object
40  CentralAir         1412 non-null   object
41  Electrical         1412 non-null   object
42  1stFlrSF           1412 non-null   int64
43  2ndFlrSF           1412 non-null   int64
44  LowQualFinSF       1412 non-null   int64

```

```

44 LowQualFinSF      1412 non-null int64
45 GrLivArea         1412 non-null int64
46 BsmtFullBath      1412 non-null int64
47 BsmtHalfBath      1412 non-null int64
48 FullBath          1412 non-null int64
49 HalfBath          1412 non-null int64
50 BedroomAbvGr      1412 non-null int64
51 KitchenAbvGr      1412 non-null int64
52 KitchenQual        1412 non-null object
53 TotRmsAbvGrd      1412 non-null int64
54 Functional         1412 non-null object
55 Fireplaces         1412 non-null int64
56 FireplaceQu        1412 non-null object
57 GarageType         1412 non-null object
58 GarageYrBlt        1412 non-null float64
59 GarageFinish       1412 non-null object
60 GarageCars         1412 non-null int64
61 GarageArea         1412 non-null int64
62 GarageQual         1412 non-null object
63 GarageCond         1412 non-null object
64 PavedDrive         1412 non-null object
65 WoodDeckSF         1412 non-null int64
66 OpenPorchSF        1412 non-null int64
67 EnclosedPorch      1412 non-null int64
68 3SsnPorch          1412 non-null int64
69 ScreenPorch        1412 non-null int64
70 PoolArea           1412 non-null int64
71 PoolQC             1412 non-null object
72 Fence              1412 non-null object
73 MiscFeature         1412 non-null object
74 MiscVal             1412 non-null int64
75 MoSold              1412 non-null int64
76 YrSold              1412 non-null int64
77 SaleType            1412 non-null object
78 SaleCondition       1412 non-null object
79 SalePrice           1412 non-null int64
80 HouseStyle_Encoded 1412 non-null int64
81 RoofStyle_Flat      1412 non-null bool
82 RoofStyle_Gable     1412 non-null bool
83 RoofStyle_Gambrel   1412 non-null bool
84 RoofStyle_Hip       1412 non-null bool
85 RoofStyle_Mansard   1412 non-null bool
86 RoofStyle_Shed      1412 non-null bool
87 Age                1412 non-null int64
88 Expensiveness       1412 non-null object
dtypes: bool(6), float64(3), int64(37), object(43)
memory usage: 934.9+ KB

```

```
In [25]: full_df=pd.concat([small_df_marked, big_df], ignore_index=True)
full_df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1426 entries, 0 to 1425
Data columns (total 95 columns):
#   Column                Non-Null Count  Dtype
---  -
0   1stFlrSF               1426 non-null  int64
1   2ndFlrSF               1426 non-null  int64
2   3SsnPorch              1426 non-null  int64
3   Age                    1426 non-null  int64
4   Alley                  1426 non-null  object
5   BedroomAbvGr           1426 non-null  int64
6   BldgType                1426 non-null  object
7   BsmtCond                1426 non-null  object
8   BsmtExposure            1426 non-null  object
9   BsmtFinSF1              1426 non-null  int64
10  BsmtFinSF2              1426 non-null  int64
11  BsmtFinType1            1426 non-null  object
12  BsmtFinType2            1426 non-null  object
13  BsmtFullBath            1426 non-null  int64
14  BsmtHalfBath            1426 non-null  int64
15  BsmtQual                1426 non-null  object
16  BsmtUnfSF               1426 non-null  int64
17  CentralAir              1426 non-null  object
18  Condition1              1426 non-null  object
19  Condition2              1426 non-null  object
20  Electrical              1426 non-null  object

```

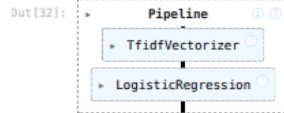
22	ExterCond	1426	non-null	object
23	ExterQual	1426	non-null	object
24	Exterior1st	1426	non-null	object
25	Exterior2nd	1426	non-null	object
26	Fence	1426	non-null	object
27	FireplaceQu	1426	non-null	object
28	Fireplaces	1426	non-null	int64
29	Foundation	1426	non-null	object
30	FullBath	1426	non-null	int64
31	Functional	1426	non-null	object
32	GarageArea	1426	non-null	int64
33	GarageCars	1426	non-null	int64
34	GarageCond	1426	non-null	object
35	GarageFinish	1426	non-null	object
36	GarageQual	1426	non-null	object
37	GarageType	1426	non-null	object
38	GarageYrBlt	1426	non-null	float64
39	GrLivArea	1426	non-null	int64
40	HalfBath	1426	non-null	int64
41	Heating	1426	non-null	object
42	HeatingQC	1426	non-null	object
43	HouseStyle	1426	non-null	object
44	HouseStyle_Encoded	1426	non-null	int64
45	Id	1426	non-null	int64
46	KitchenAbvGr	1426	non-null	int64
47	KitchenQual	1426	non-null	object
48	LandContour	1426	non-null	object
49	LandSlope	1426	non-null	object
50	LotArea	1426	non-null	int64
51	LotConfig	1426	non-null	object
52	LotFrontage	1426	non-null	float64
53	LotShape	1426	non-null	object
54	LowQualFinSF	1426	non-null	int64
55	MSSubClass	1426	non-null	int64
56	MSSZoning	1426	non-null	object
57	MasVnrArea	1426	non-null	float64
58	MasVnrType	1426	non-null	object
59	MiscFeature	1426	non-null	object
60	MiscVal	1426	non-null	int64
61	MoSold	1426	non-null	int64
62	Neighborhood	1426	non-null	object
63	OpenPorchSF	1426	non-null	int64
64	OverallCond	1426	non-null	int64
65	OverallQual	1426	non-null	int64
66	PavedDrive	1426	non-null	object
67	PoolArea	1426	non-null	int64
68	PoolQC	1426	non-null	object
69	RoofMatl	1426	non-null	object
70	RoofStyle_Flat	1426	non-null	bool
71	RoofStyle_Gable	1426	non-null	bool
72	RoofStyle_Gambrel	1426	non-null	bool
73	RoofStyle_Hip	1426	non-null	bool
74	RoofStyle_Mansard	1426	non-null	bool
75	RoofStyle_Shed	1426	non-null	bool
76	SaleCondition	1426	non-null	object
77	SalePrice	1426	non-null	int64
78	SaleType	1426	non-null	object
79	ScreenPorch	1426	non-null	int64
80	Street	1426	non-null	object
81	TotRmsAbvGrd	1426	non-null	int64
82	TotalBsmtSF	1426	non-null	int64
83	Utilities	1426	non-null	object
84	WoodDeckSF	1426	non-null	int64
85	YearBuilt	1426	non-null	int64
86	YearRemodAdd	1426	non-null	int64
87	YrSold	1426	non-null	int64
88	annotation_id	14	non-null	float64
89	annotator	14	non-null	float64
90	created_at	14	non-null	object
91	id	14	non-null	float64
92	lead_time	14	non-null	float64
93	Expensiveness	1426	non-null	object
94	updated_at	14	non-null	object

dtypes: bool(6), float64(7), int64(37), object(45)  
memory usage: 1000.0+ KB

```
In [30]: X_train, X_test, y_train, y_test = train_test_split(
full_df[['LotShape']], # текст рецензии - будет использоваться как входные данные
full_df[['Expensiveness']], # метка классов (положительный или отрицательный)
test_size=0.2, # доля данных которая попадет в тест (20%)
random_state=42 # Зерно генератора случайных чисел для воспроизводимости результатов
)
```

```
In [31]: pipeline = Pipeline([
('tfidf', TfidfVectorizer(stop_words='english')), # векторизация текста, с исключением стоп слов англ языка
('clf', LogisticRegression(max_iter=1000)) # модель логистической регрессии с увеличенным количеством итераций до 1000
])
```

```
In [32]: #обучение модели на обучающем наборе данных
pipeline.fit(X_train, y_train)
```



```
In [33]: predictions = pipeline.predict(X_test) # предсказание модели на тестовых данных
accuracy = accuracy_score(y_test, predictions) # расчет точности модели
report = classification_report(y_test, predictions) # оценка качества предсказания

print(f"Точность модели {accuracy}")
print('Отчет по классификации')
print(report)
```

Точность модели 0.6853146853146853

```
Отчет по классификации
              precision    recall  f1-score   support

     Chip              0.00        0.00        0.00         21
  Expensive              0.00        0.00        0.00         69
     Medium              0.69        1.00        0.81        196

 accuracy              0.23        0.33        0.69        286
 macro avg              0.23        0.33        0.27        286
 weighted avg           0.47        0.69        0.56        286
```

```

/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/sklearn/metrics/_classification.py:1517: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, f'{metric.capitalize()} is', len(result))
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/sklearn/metrics/_classification.py:1517: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, f'{metric.capitalize()} is', len(result))
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/sklearn/metrics/_classification.py:1517: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
  _warn_prf(average, modifier, f'{metric.capitalize()} is', len(result))

```