Лекция 2

Предобработка данных

1. Введение

Предобработка данных — это неотъемлемая часть жизненного цикла машинного обучения, нацеленная на обеспечение качества, целостности и удобства обработки данных, которые будут использоваться в обучении моделей. В процессе предобработки осуществляется подготовка сырых данных для дальнейшего анализа и моделирования. Часто, данные, собранные из различных источников, содержат ошибки, пропуски и избыточную информацию, которые могут значительно снизить точность модели. В результате предобработки данные приводятся к единому формату и очищаются, обеспечивая повышение качества и производительности обучающего алгоритма.

Ключевыми этапами предобработки данных являются очистка, нормализация, стандартизация, обработка пропусков, кодирование категориальных переменных и создание новых признаков. Эти шаги помогают улучшить эффективность модели, устраняя потенциальные ошибки, которые могут возникнуть из-за некачественных данных. Далее мы рассмотрим каждый из этапов более подробно.

1. Очистка данных

Очистка данных — это ключевой процесс, направленный на выявление и устранение проблемных значений, а также на исправление возможных ошибок и аномалий. Этот этап является особенно важным, поскольку наличие некорректных данных может привести к существенным отклонениям в результатах моделирования и снижению качества предсказаний. Ошибки в данных могут возникнуть по ряду причин, включая человеческие ошибки, сбои оборудования или программные сбои в процессе сбора и преобразования данных. В связи с этим очистка данных становится основой для создания достоверных и устойчивых моделей.

Этапы очистки данных обычно включают следующие ключевые шаги:

1. **Обнаружение и обработка пропущенных значений.** Пропущенные значения могут значительно влиять на производительность модели. Наиболее распространенные подходы к обработке пропусков включают удаление строк с пропущенными значениями, замену пропусков средними или медианными значениями признаков, а также применение

более сложных методов, таких как метод ближайших соседей или предсказание значений на основе регрессионных моделей.

2. **Исправление ошибок форматирования и типов данных.** Ошибки в формате данных, такие как использование разных единиц измерения, неверный формат дат или смешение типов данных, могут повлиять на корректность анализа. На этапе очистки данных важно привести все данные к единому формату. Например, даты могут быть конвертированы в формат, удобный для дальнейшего анализа, а текстовые данные — в числовые или категориальные значения.
3. **Обнаружение и устранение выбросов.** Выбросы — это значения, которые существенно отличаются от других наблюдений в наборе данных и могут возникать из-за ошибок при измерениях или уникальных событий, не характерных для исследуемого процесса. Обнаружение выбросов может осуществляться с помощью визуализации (например, с использованием диаграммы размаха) или статистических методов (например, на основе интерквартильного размаха или Z-оценок). В зависимости от задачи выбросы могут быть либо удалены, либо обработаны специальным образом, например, заменены средними значениями или сглажены.
4. **Обработка дубликатов.** В случае наличия дубликатов в данных, особенно если речь идет о больших наборах данных, это может привести к необоснованному увеличению веса определенных наблюдений и искажению результатов модели. На этапе очистки проводится проверка на наличие дублирующихся записей и их удаление.
5. **Коррекция значений.** Иногда значения в данных могут быть неправдоподобными (например, возраст более 150 лет или отрицательные значения роста), и их следует корректировать или удалять. В случае корректировки необходимо использовать логически оправданные подходы для замены недопустимых значений.

Эти этапы очистки данных создают более качественную основу для моделирования, что помогает снизить ошибки в прогнозах и улучшить общее качество работы алгоритма машинного обучения.

2. Основные шаги предобработки данных

Предобработка включает в себя множество шагов, каждый из которых служит определенной цели в подготовке данных. Ниже представлены основные этапы, которые необходимо рассмотреть.

- **Импорт данных**: Первый шаг в любом проекте машинного обучения – сбор и загрузка данных из различных источников, таких как файлы CSV, базы данных или даже большие наборы данных в реальном времени.
- **Обработка пропущенных значений**: Один из самых распространенных вызовов – это управление пропущенными значениями. Их можно заменить средним, медианным значением или наиболее часто встречающимся значением в столбце, или же использовать более сложные методы, такие как предсказание пропущенных значений с помощью других характеристик.
- **Кодирование категориальных данных**: Большинство алгоритмов машинного обучения требуют ввода в числовом формате. Категориальные данные нужно преобразовать в числовые, используя методы, такие как One-Hot Encoding или Label Encoding.
- **Масштабирование признаков**: Различия в масштабах признаков могут негативно повлиять на производительность некоторых алгоритмов. Масштабирование признаков включает в себя стандартизацию или нормализацию данных.
- **Разделение данных на обучающие и тестовые выборки**: Важно разделить данные на две части: одну для обучения модели и другую для тестирования ее производительности. Это помогает избежать переобучения и лучше оценить, как модель будет работать на новых данных.

3. Обработка пропущенных значений

Пропущенные данные могут быть результатом множества факторов, включая ошибки ввода, потери данных при передаче или недоступность информации. Обработка пропущенных значений требует внимательного анализа:

- **Удаление**: Простой, но иногда рискованный метод. Удаление строк или столбцов, содержащих пропущенные значения, может привести к потере важной информации.
- **Импутация**: Техники импутации заменяют пропущенные значения на основе других данных. Например, можно использовать среднее или медианное значение признака для числовых данных или модальное значение для категориальных данных. Более сложные методы включают использование моделей машинного обучения для предсказания пропущенных значений.

4. Кодирование категориальных данных

Категориальные данные часто представляют собой переменные, которые содержат метки, не имеющие математического значения. Примеры включают пол, национальность или категории продуктов. Существует несколько подходов для преобразования этих данных:

- **One-Hot Encoding**: Создает новый столбец для каждого уникального значения в категориальной переменной. Каждый столбец будет содержать 0 или 1, соответствующие отсутствию или присутствию значения в данной строке.
- **Label Encoding**: Присваивает каждой категории уникальное число. Хотя этот метод эффективен, он может ввести элемент порядка там, где его нет.

5. Масштабирование признаков

Масштабирование признаков помогает нормализовать данные в рамках определенного диапазона и убирает проблемы, связанные с искусственно высоким весом более крупных значений. Существуют различные методы масштабирования:

- **Стандартизация**: Преобразует признаки, вычитая среднее значение и делит на стандартное отклонение, чтобы добиться среднего 0 и дисперсии 1.
- **Нормализация**: Масштабирует данные на интервал от 0 до 1 или от -1 до 1.

6. Разделение данных

Финальный этап предобработки данных — разделение их на обучающий, тестовый и иногда валидационный наборы, что необходимо для объективной оценки производительности модели. Основная цель этого этапа — подготовить данные так, чтобы модель могла обучаться на одном наборе и проходить тестирование на другом, не содержащем информации из обучающего набора. Такой подход помогает избежать переобучения и проверить, насколько хорошо модель будет справляться с новыми данными, с которыми она ранее не сталкивалась.

Стандартное соотношение для разделения данных — 80/20 или 70/30, где 80% или 70% данных используются для обучения, а 20% или 30% — для тестирования. Обучающий набор используется для оптимизации параметров модели, чтобы она могла максимально точно отражать закономерности в данных. Тестовый набор, напротив, используется только для оценки модели после завершения процесса обучения. Он позволяет оценить способность модели к генерализации, т.е. её способность правильно обрабатывать новые данные, с которыми она не сталкивалась в процессе обучения.

Часто данные делятся и на три части, выделяя помимо обучающего и тестового наборов также валидационный набор. Валидационный набор позволяет настроить гиперпараметры модели, такие как коэффициенты регуляризации, количество слоев и узлов в нейронной сети или количество деревьев в ансамблевых методах. Это также предотвращает утечку информации из

тестового набора в процесс настройки модели, что могло бы искусственно завысить ее точность на тестовых данных.

Существует несколько подходов к разделению данных:

1. **Случайное разделение.** Один из простых способов заключается в случайном делении данных на обучающий и тестовый наборы. Однако для небольших выборок случайное разделение может привести к неравномерному распределению классов или признаков, что исказит результаты.
2. **Стратифицированное разделение.** Этот подход особенно полезен для несбалансированных данных, когда одна категория или класс встречается значительно чаще других. Стратифицированное разделение сохраняет пропорцию классов в обучающем и тестовом наборах, что позволяет избежать перекоса данных и сделать результаты тестирования более надежными.
3. **K-кратная перекрестная проверка (k-fold cross-validation).** Этот метод особенно эффективен для небольших выборок и предполагает многократное разделение данных на обучающие и тестовые части. Данные делятся на k частей, и каждая часть поочередно используется как тестовый набор, а оставшиеся части — как обучающий. Результаты по всем итерациям усредняются, что позволяет получить более точную оценку производительности модели.

Правильное разделение данных позволяет объективно оценить модель, минимизируя риск переобучения и искажений, связанных с определенными структурами в данных. Этот процесс закладывает прочную основу для последующих этапов машинного обучения, обеспечивая надежные и реалистичные результаты.

7. Заключение

Предобработка данных – это фундаментальный этап в любом процессе анализа данных и машинного обучения. От тщательности выполнения этого этапа зависит успех всего проекта. Качественная предобработка повышает вероятность того, что модель машинного обучения будет работать эффективно и точно.