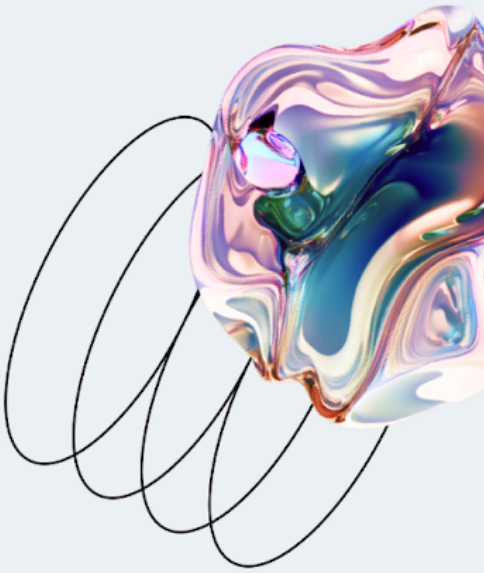
GeekBrains



Аналитика данных и ETL

Data Science

Оглавление

Введение

Всем привет! Мы начинаем наш урок по теме аналитика данных и ETL, где обсудим:

- Что такое бизнес-аналитика и для чего она нужна.
- Что такое анализ данных, из каких этапов он состоит и для чего нужен.

После того как разберёмся с анализом данных, мы рассмотрим, как применять python для задач связанных с анализом данных, точнее, как производить обработку данных, используя python. В последней части урока поговорим о ETL процессах, обсудим, что это такое и почему важно для дата-сайнтиста. Поговорим о том, как извлекать данные, трансформировать их и партицировать, то есть разбивать большие данные на логические части по выбранным критериям.

Словарь терминов

ETL — аббревиатура от Extract, Transform, Load. Это общий термин для процессов, которые происходят, когда данные переносят из нескольких систем в одно хранилище.

Модель данных — это совокупность структур данных и операций их обработки.

Атрибут — свойство некоторой сущности. Часто называется полем таблицы.

Домен атрибута — множество допустимых значений, которые может принимать атрибут.

Кортеж — конечное множество взаимосвязанных допустимых значений атрибутов, которые вместе описывают некоторую сущность (строка таблицы).

Отношение — конечное множество кортежей (таблица).

Схема отношения — конечное множество атрибутов, определяющих некоторую сущность. Иными словами, это структура таблицы, состоящей из конкретного набора полей.

Проекция — отношение, полученное из заданных исходных данных путём удаления и (или) перестановки некоторых атрибутов.

Функциональная зависимость между атрибутами (множествами атрибутов) X и Y означает, что для любого допустимого набора кортежей в этом отношении: если два кортежа совпадают по значению X, то они совпадают по значению Y. Например, если значение атрибута «Название компании» — Canonical Ltd, то значением

атрибута «Штаб-квартира» в таком кортеже всегда будет Millbank Tower, London, United Kingdom. Обозначение: {X} → {Y}.

Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц).

Метод нормальных форм (НФ) состоит в сборе информации об объектах решения задачи в рамках одного отношения и последующей декомпозиции этого отношения на несколько взаимосвязанных отношений на основе процедур нормализации отношений.

Цель нормализации — исключить избыточное дублирование данных, которое является причиной аномалий, возникших при добавлении, редактировании и удалении кортежей (строк таблицы).

Аномалией называется ситуация в таблице БД, приводящая к противоречию в БД или существенно усложняющая обработку БД. Причиной является излишнее дублирование данных в таблице, которое вызывается наличием функциональных зависимостей от неключевых атрибутов.

Аномалии-модификации проявляются в том, что изменение одних данных может повлечь просмотр всей таблицы и соответствующее изменение некоторых записей.

Аномалии-удаления — при удалении какого-либо кортежа из таблицы может пропасть информация, которая не связана напрямую с удаляемой записью.

Аномалии-добавления возникают, когда информацию в таблицу нельзя поместить, пока она не полная, либо вставка записи требует дополнительного просмотра таблицы.

Бизнес аналитика и анализ данных

Обсудим, что такое бизнес-аналитика и для чего она нужна компаниям. **Бизнес-аналитика** (BI — Business intelligence) — это набор процессов, архитектур и технологий, которые преобразуют необработанные данные в значимую информацию, стимулирующую прибыльные бизнес-действия. Это набор программ и услуг, которые преобразуют данные в действенный интеллект и знания.

Бизнес-аналитика напрямую влияет на стратегические, тактические и оперативные бизнес-решения организации, а также поддерживает принятие решений на основе фактов, используя исторические данные, а не предположения и интуицию.

Инструменты, которыми пользуется бизнес-аналитика, выполняют анализ данных и создают отчёты, сводки, информационные панели, карты, графики и диаграммы, чтобы предоставить пользователям подробные сведения о характере бизнеса.

Итак, выделим несколько пунктов почему бизнес-аналитика имеет важное значение:

- Измерение: создание KPI (ключевых показателей эффективности) на основе исторических данных.
- Определите и установите критерии для различных процессов.
- С помощью BI-систем организации могут выявлять рыночные тенденции и выявлять бизнес-проблемы, которые необходимо решить.
- BI помогает визуализировать данные, что повышает качество данных и, следовательно, качество принятия решений.
- Системы BI могут использоваться не только предприятиями, но и МСП (малые и средние предприятия).

Рассмотрим несколько примеров применения бизнес-аналитики в различных сферах:

Пример 1:

Владелец отеля использует аналитические приложения BI для сбора статистической информации о средней загруженности и стоимости номера. Это помогает найти совокупный доход за номер.

Он также собирает статистические данные о доле рынка и данные опросов клиентов каждого отеля, чтобы определить его конкурентную позицию на различных рынках.

Анализируя эти тенденции из года в год, месяц за месяцем и день за днём помогает руководству предлагать скидки на аренду помещений.

Пример 2:

Банк предоставляет менеджерам филиалов доступ к BI-приложениям. Это помогает менеджеру филиала определить, кто является наиболее прибыльным клиентом и с какими клиентами им следует работать.

Использование инструментов BI освобождает сотрудников информационных технологий от задачи составления аналитических отчётов для отделов. Это также даёт персоналу отдела доступ к более богатым источникам данных.

Тенденции в бизнес-аналитике

Ниже приведены некоторые тенденции бизнес-аналитики и аналитики, о которых вам следует знать.

Искусственный интеллект: отчёт аналитической компании Gartner указывает, что ИИ и машинное обучение теперь выполняют сложные задачи, выполняемые человеческим интеллектом. Эта возможность используется для анализа данных в реальном времени и создания отчётов на приборной панели.

Collaborative BI: программное обеспечение BI в сочетании с инструментами совместной работы, в том числе социальными сетями, и другими новейшими технологиями расширяют возможности коллективной работы и обмена информацией для совместного принятия решений.

Embedded BI: Embedded BI позволяет интегрировать программное обеспечение BI или некоторые его функции в другое бизнес-приложение для расширения и расширения его функций отчётности.

Облачная аналитика: BI-приложения скоро будут предлагаться в облаке, и всё больше компаний будут переходить на эту технологию. Согласно их прогнозам, в течение пары лет расходы на облачную аналитику будут расти в 4,5 раза быстрее.

Подведём итоги и ответим на вопрос:

Вопрос: Какое назначение у BI? Какие плюсы и минусы вы видите у систем бизнес-аналитики?

BI-системы помогают предприятиям определять тенденции рынка и выявлять проблемы бизнеса, которые необходимо решить.

BI-технология может использоваться аналитиком данных, ИТ-специалистами, бизнес-пользователями и главой компании.

Система BI помогает организации улучшить видимость, производительность и исправить отчётность.

Недостатки BI в том, что это трудоёмкий, дорогостоящий и очень сложный процесс.

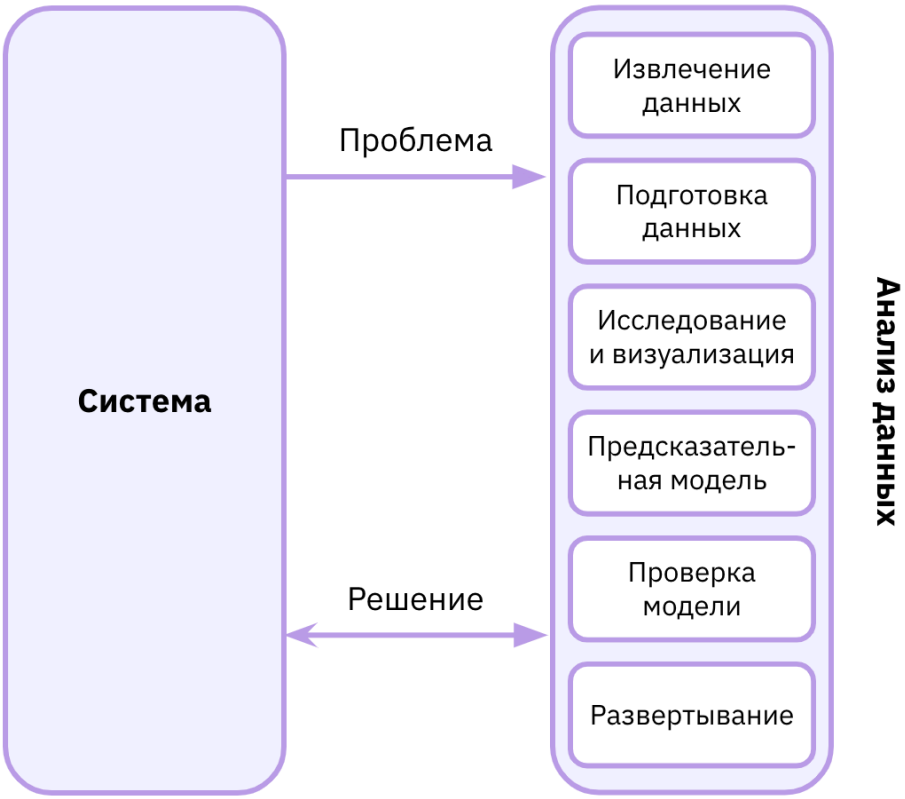
Анализ данных

Анализ данных можно описать как процесс, состоящий из нескольких шагов, в которых сырые данные превращаются и обрабатываются с целью создать визуализации и сделать предсказания на основе математической модели.

Анализ данных — это всего лишь последовательность шагов, каждый из которых играет ключевую роль для последующих. Этот процесс похож на цепь последовательных, связанных между собой этапов:

- Определение проблемы.
- Извлечение данных.
- Подготовка данных — очистка данных.
- Подготовка данных — преобразование данных.
- Исследование и визуализация данных.
- Моделирование.
- Оценка (проверка) модели.
- Развёртывание — визуализация и интерпретация результатов.
- Развёртывание — развёртывание решения.

На схеме ниже приведён процесс анализа данных:



Подробнее изучим процесс анализа данных.

Понимание бизнеса

На этом этапе устанавливаются цели бизнеса и добычи данных.

- Вы должны понимать цели бизнеса и клиента. Вы должны определить, что хочет ваш клиент (что часто даже они сами не знают).
- Подведите итоги текущего сценария добычи данных. Фактор ресурсов, предположения, ограничения и другие существенные факторы в вашей оценке.
- Используя бизнес-цели и текущий сценарий, определите свои цели интеллектуального анализа данных.
- Хороший план интеллектуального анализа данных очень подробный и должен быть разработан для достижения целей как бизнеса, так и данных.

Понимание данных

На этом этапе выполняется проверка работоспособности данных, чтобы проверить, подходит ли она для целей интеллектуального анализа данных.

- Данные собираются из нескольких источников данных, доступных в организации.

- Эти источники данных могут включать в себя несколько баз данных, файлы или хранилища данных. Существуют такие проблемы, как сопоставление объектов и интеграция схем, которые могут возникнуть в процессе интеграции данных. Это довольно сложный процесс, так как данные из разных источников вряд ли легко сопоставимы. Например, таблица A содержит сущность с именем номер клиента, тогда как другая таблица B содержит сущность с именем идентификационный номер клиента.

- Следовательно, довольно трудно гарантировать, что оба эти заданных объекта ссылаются на одно и то же значение или нет. Здесь метаданные должны использоваться для уменьшения ошибок в процессе интеграции данных.

- Далее необходимо выполнить поиск свойств полученных данных. Хороший способ исследовать данные — это ответить на вопросы интеллектуального анализа данных (решённые в бизнес-фазе), используя инструменты запросов, отчётов и визуализации.

- На основании результатов запроса должно быть установлено качество данных. Отсутствующие данные, если таковые имеются, должны быть получены.

Подготовка данных

На этом этапе данные готовятся к производству. Процесс подготовки данных занимает около 90% времени проекта.

Данные из разных источников должны быть отобраны, очищены, преобразованы, отформатированы и анонимизированы.

Очистка данных — это процесс «очистки» данных путём сглаживания зашумлённых данных, такие данные повреждены или искажены, и заполнения пропущенных значений.

Например, для демографического профиля клиента отсутствуют данные о возрасте. Данные являются неполными и должны быть заполнены. В некоторых случаях могут быть выбросы данных. Например, возраст имеет значение 300. Данные могут быть противоречивыми. Например, имя клиента отличается в разных таблицах.

Операции преобразования данных изменяют данные, чтобы сделать их полезными для интеллектуального анализа данных.

Преобразование данных

Операции преобразования данных будут способствовать успеху процесса майнинга.

Сглаживание: помогает удалить шум из данных.

Агрегация. Сводные или агрегирующие операции применяются к данным. То есть еженедельные данные о продажах агрегируются для расчёта месячной и годовой суммы.

Обобщение: На этом этапе данные низкого уровня заменяются концепциями более высокого уровня с помощью иерархий концепций. Например, город заменяется графством.

Нормализация: Нормализация выполняется, когда данные атрибута увеличиваются или уменьшаются. Пример: данные должны находиться в диапазоне от -2,0 до 2,0 после нормализации.

Построение атрибута: эти атрибуты создаются и включают в себя заданный набор атрибутов, полезных для интеллектуального анализа данных.

Результатом этого процесса является окончательный набор данных, который можно использовать при моделировании.

Исследование и визуализация данных

Изучение данных — это их анализ в графической или статистической репрезентации с целью поиска моделей или взаимосвязей. **Визуализация — лучший инструмент для выделения подобных моделей.**

За последние годы визуализация данных развилась так сильно, что стала независимой дисциплиной. Многочисленные технологии используются исключительно для отображения данных, а многие типы отображения работают так, чтобы получать только лучшую информацию из набора данных.

Исследование данных состоит из предварительного изучения, которое необходимо для понимания типа и значения собранной информации. Вместе с информацией, собранной при определении проблемы, такая категоризация определяет, какой метод анализа данных лучше всего подойдёт для определения модели.

Эта фаза, в дополнение к изучению графиков, состоит из следующих шагов:

- Обобщение данных.
- Группировка данных.

- Исследование отношений между разными атрибутами.
- Определение моделей и тенденций.
- Построение моделей регрессионного анализа.
- Построение моделей классификации.

Как правило, анализ данных требует обобщения заявлений касательно изучаемых данных.

Моделирование

Предсказательная аналитика — это процесс в анализе данных, который нужен для создания или поиска подходящей статистической модели для предсказания вероятности результата.

После изучения данных у вас есть вся необходимая информация для развития математической модели, которая кодирует отношения между данными. Эти модели полезны для понимания изучаемой системы и используются в двух направлениях.

Первое — предсказания о значениях данных, которые создаёт система. В этом случае речь идёт о регрессионных моделях.

Второе — классификация новых продуктов. Это уже модели классификации или модели кластерного анализа.

Простые методы генерации этих моделей включают такие техники:

- Линейная регрессия.
- Логистическая регрессия.
- Классификация.
- Дерево решений.
- Метод k-ближайших соседей.

Но таких методов много, и у каждого есть свои характеристики, которые делают их подходящими для определённых типов данных и анализа. Каждый из них приводит к появлению определённой модели, а их выбор соответствует природе модели продукта.

Некоторые из методов будут предоставлять значения, относящиеся к реальной системе и их структурам. Они смогут объяснить некоторые характеристики изучаемой системы простым способом. Другие будут делать хорошие предсказания,

но их структура будет оставаться «чёрным ящиком» с ограниченной способностью объяснить характеристики системы.

Оценка

На этом этапе идентифицированные шаблоны оцениваются в соответствии с бизнес-целями.

- Результаты, полученные с помощью модели интеллектуального анализа данных, должны оцениваться в соответствии с бизнес-целями.
- Получение понимания бизнеса является итеративным процессом. Фактически, при понимании, новые бизнес-требования могут быть повышены из-за интеллектуального анализа данных.
- Принято решение о переходе модели на этап развёртывания.

Развёртывание

На этапе развёртывания вы отправляете свои открытия для интеллектуального анализа данных в повседневные бизнес-операции.

- Знания или информация, обнаруженные в процессе извлечения данных, должны быть понятны для нетехнических заинтересованных сторон.
- Создан подробный план развёртывания для доставки, обслуживания и мониторинга обнаружений интеллектуального анализа данных.
- Окончательный отчёт по проекту создаётся с учётом извлечённых уроков и ключевых событий в ходе проекта. Это помогает улучшить деловую политику организации.

Python для базовых задач обработки данных

Что такое предварительная обработка данных и зачем она нам нужна?

Чтобы алгоритмы машинного обучения работали, необходимо преобразовать **необработанные данные** в **чистый набор данных** , что означает, что мы должны преобразовать набор данных в **числовые данные** . Мы делаем это, кодируя все **категориальные метки** в векторы-столбцы с двоичными значениями. **Отсутствующие значения** или NaN (не число) в наборе данных являются раздражающей проблемой. Вы должны либо удалить недостающие

строки, либо заполнить их средними или интерполированными значениями. Для примера возьмём открытый дата сет с данными о катастрофе Титаника.

Для работы с данными вы можете загрузить CSV в Pandas. Для этого сделаем импорт библиотек и прочитаем файл.

```
import pandas as pd
import numpy as np
df = pd.read_csv('train_ds2.csv')
```

Посмотрим на формат данных ниже:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Если вы внимательно посмотрите на приведённую выше сводку Pandas, всего 891 строка, но колонка Age содержит только 714 (что означает, что нам не хватает некоторых данных), в колонке Embarked отсутствуют две строки и в колонке Cabin отсутствует много данных. Типы данных объекта не являются числовыми, поэтому нам нужно найти способ перекодировать их в числовые значения, так как многие модели машинного обучения умеют работать только с числовым представлением данных.

Отбросьте бесполезные столбцы

Попробуем удалить некоторые из столбцов, которые не будут иметь большого значения для нашей модели машинного обучения. Начнём с Name, Ticket и Cabin.

```
cols = ['Name', 'Ticket', 'Cabin']
df = df.drop(cols, axis=1)
```

Мы удалили три столбца:

```
df.info()

RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Sex          891 non-null    object
 4   Age          714 non-null    float64
 5   SibSp        891 non-null    int64
 6   Parch        891 non-null    int64
 7   Fare         891 non-null    float64
 8   Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(2)
memory usage: 62.8+ KB
```

Отбросьте строки с пропущенными значениями

Затем мы можем удалить все строки данных, в которых есть пропущенные значения (NaN). Вот как:

```
df.dropna()
```

```
df.info()
```

```
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Sex          891 non-null    object
 4   Age          714 non-null    float64
 5   SibSp        891 non-null    int64
 6   Parch        891 non-null    int64
 7   Fare         891 non-null    float64
 8   Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(2)
memory usage: 62.8+ KB
```

Кроме этого значения с NaN можно не только удалять но и заменять их каким-то определённым значением (чаще всего это константа представляющая собой медианное значение, но всё зависит от конкретной ситуации) используя функцию filna().

Проблема с удалением строк

После удаления строк с отсутствующими значениями мы обнаруживаем, что набор данных сократился до 712 строк с 891, что означает, что мы тратим **данные впустую** . Модели машинного обучения нуждаются в данных для обучения и хорошей работы. Итак, сохраним данные и будем использовать их как можно чаще. Подробнее об этом ниже.

Создание фиктивных переменных

Вместо того, чтобы тратить наши данные впустую, преобразуем столбцы Pclass, Sex и Embarked в Pandas дата фрейм и отбросим их после преобразования.

```
1 dummies = []
2 cols = ['Pclass', 'Sex', 'Embarked']
3 for col in cols:
4     dummies.append(pd.get_dummies(df[col]))
```

Преобразуем полученный массив в дата фрейм.

```
1 titanic_dummies = pd.concat(dummies, axis=1)
```

Теперь, после преобразования, у нас есть 8 колонок, где колонки 1, 2 и 3 представляют собой класс билета конкретного пассажира. Присоединим их к исходному дата фрейму.

```
df = pd.concat((df,titanic_dummies), axis=1)
```

После преобразования удалим ненужные столбцы, преобразованные данные из которых мы уже присоединили.

```
df = df.drop(['Pclass', 'Sex', 'Embarked'], axis=1)
```

Посмотрим на новый фрейм данных:

```
df.info()
```

```
 1   Survived     891 non-null    int64
 2   Age          714 non-null    float64
 3   SibSp        891 non-null    int64
 4   Parch        891 non-null    int64
 5   Fare         891 non-null    float64
 6   1            891 non-null    uint8
 7   2            891 non-null    uint8
 8   3            891 non-null    uint8
 9   female       891 non-null    uint8
 10  male         891 non-null    uint8
 11  C            891 non-null    uint8
 12  Q            891 non-null    uint8
 13  S            891 non-null    uint8
dtypes: float64(2), int64(4), uint8(8)
memory usage: 48.9 KB
```

Позаботьтесь о недостающих данных

Теперь всё чисто, кроме колонки Age, в которой много пропущенных значений. Вычислим медиану или интерполируем все возрасты и заполним эти недостающие значения возраста. В Pandas есть функция interpolate(), которая заменит все отсутствующие значения NaN интерполированными значениями.

```
df.info()
```

```
  1   Survived     891 non-null    int64
  2   Age          891 non-null    float64
  3   SibSp        891 non-null    int64
  4   Parch        891 non-null    int64
  5   Fare         891 non-null    float64
  6   1            891 non-null    uint8
  7   2            891 non-null    uint8
  8   3            891 non-null    uint8
  9   female       891 non-null    uint8
  10  male         891 non-null    uint8
  11  C            891 non-null    uint8
  12  Q            891 non-null    uint8
  13  S            891 non-null    uint8
dtypes: float64(2), int64(4), uint8(8)
memory usage: 48.9 KB
```

Преобразуйте фрейм данных в NumPy

Теперь, когда мы преобразовали все данные в целые числа, пришло время подготовить данные для моделей машинного обучения. Здесь в игру вступают scikit-learn и NumPy:

X= Входной набор с 14 атрибутами (подготовленный нами ранее df).

y = Небольшой результирующий выход (значения которые мы хотим предсказывать) в данном случае колонка Survived.

Теперь мы преобразуем наш фрейм данных из Pandas в NumPy и назначаим входные и выходные значения:

```
1 X = df.values
2 y = df['Survived'].values
```

Так как в нашем входном массиве всё ещё присутствует колонка Survived, которой быть не должно, удалим её. Мы знаем что она была первой в нашем датафрейме и поэтому её индекс в массиве будет равен 1.

```
1 X = np.delete(X, 1, axis=1)
```

Разделите набор данных на обучающие данные и тестовые данные

Теперь, когда у нас готовы входные и выходные данные, разделим набор данных: мы выделим 70 процентов для обучения и 30 процентов для тестов с использованием модуля scikit-learn.

```
1 from sklearn.model_selection import train_test_split
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

На этом всё, что касается базовой предобработки данных. Теперь вы можете предварительно обрабатывать данные самостоятельно и мы попробуем сделать это на нашем семинаре.

Основные функции ETL-систем

Система ETL — один из центральных процессов в системах хранения данных. Он включает в себя:

- **Извлечение** данных из различных источников.
- **Трансформация** и очистка данных для приведения их к единообразию или в соответствие с бизнес-задачами.
- **Загрузка** в хранилище данных.

По мере роста популярности баз данных в 1970-х годах ETL был представлен как процесс интеграции и загрузки данных для вычислений и анализа, который в итоге стал основным методом обработки данных для проектов хранилища данных.

ETL обеспечивает основу для анализа данных и рабочих потоков машинного обучения. С помощью ряда бизнес-правил ETL очищает и организует данные таким образом, чтобы удовлетворить конкретные потребности бизнес-аналитики, такие как ежемесячная отчётность, но также может выполнять более сложную аналитику, которая может улучшить внутренние процессы или взаимодействие с конечными пользователями. ETL часто используется организацией для:

- Извлечение данных из устаревших систем.
- Очистите данные, чтобы улучшить качество данных и обеспечить согласованность.
- Загрузить данные в целевую базу данных.

Как работает ETL

Самый простой способ понять, как работает ETL, — понять, что происходит на каждом этапе процесса.

Извлечение

Во время извлечения данных необработанные данные копируются или экспортируются из исходных местоположений в промежуточную область. Группы управления данными могут извлекать данные из различных источников данных, которые могут быть структурированными или неструктурированными. Эти источники включают, но не ограничиваются:

- SQL- или NoSQL- серверы.
- CRM и ERP системы управления клиентами и процессами.
- Плоские файлы, в которых данные хранятся в виде обычного текста, часто в виде таблицы со строками и столбцами.
- Электронный адрес.
- Интернет-страницы.

Трансформация

В промежуточной области необработанные данные подвергаются обработке. Здесь данные преобразуются и консолидируются для предполагаемого аналитического использования. Этот этап может включать в себя следующие задачи:

- Фильтрация, очистка, дедупликация (удаление дупликатов), проверка и аутентификация (проверки подлинности) данных.
- Выполнение расчётов, переводов или суммирования на основе необработанных данных. Это может включать изменение заголовков строк и столбцов для согласованности, конвертацию валют или других единиц измерения, редактирование текстовых строк и многое другое.
- Проведение аудитов для обеспечения качества данных и соответствия.
- Удаление, шифрование или защита данных регулируется отраслевыми или государственными регулирующими органами.
- Форматирование данных в таблицы или объединённые таблицы в соответствии со схемой целевого хранилища данных.

Загрузка

На этом последнем шаге преобразованные данные перемещаются из промежуточной области в целевое хранилище данных. Как правило, это включает первоначальную загрузку всех данных с последующей периодической загрузкой добавочных изменений данных и, реже, полным обновлением для удаления и замены данных в хранилище.

Для большинства организаций, использующих ETL, этот процесс является автоматизированным, чётко определённым, непрерывным и управляемым пакетами. Обычно ETL выполняется в нерабочее время, когда трафик в исходных системах и хранилище данных минимален.

Проблема, из-за которой в принципе родилась необходимость использовать решения ETL, заключается в потребностях бизнеса в получении достоверной отчётности из того бардака, который творится в данных любой ERP-системы.

Этот бардак есть всегда, он бывает двух видов:

1. Случайные ошибки, возникшие на уровне ввода, переноса данных, или из-за багов.
2. Различия в справочниках и детализации данных между смежными ИТ-системами.

При этом если первый вид бардака побороть можно, то второй вид по большей части не является ошибкой — контролируемые различия в структуре данных, это нормальная оптимизация под цели конкретной системы.

Из-за этой особенности ETL-системы должны в идеале решать не одну, а две задачи:

1. Привести все данные к единой системе значений и детализации, попутно обеспечив их качество и надёжность.
2. Обеспечить аудиторский след при преобразовании (Transform) данных, чтобы после преобразования можно было понять, из каких именно исходных данных и сумм собралась каждая строчка преобразованных данных.

Как работает ETL система

Все основные функции ETL системы умещаются в следующий процесс:

В разрезе потока данных это несколько систем-источников (обычно OLTP — система транзакций) и система приёмник (обычно OLAP — аналитическая система), а также пять стадий преобразования между ними:

1. Процесс загрузки — Его задача затянуть в ETL данные произвольного качества для дальнейшей обработки, на этом этапе важно сверить суммы пришедших строк, если в исходной системе больше строк, чем в RawData (сырые данные — данные полученные из источника без проведения анализа и очистки) то значит — загрузка прошла с ошибкой.

2. Процесс валидации данных — на этом этапе данные последовательно проверяются на корректность и полноту, составляется отчёт об ошибках для исправления.

3. Процесс мэппинга (сопоставление, приведение к общей структуре) данных с целевой моделью — на этом этапе к валидированной таблице пристраивается ещё n-столбцов по количеству справочников целевой модели данных, а потом по таблицам мэппингов в каждой пристроенной ячейке, в каждой строке проставляются значения целевых справочников. Значения могут проставляться как 1:1, так и *:1, так и 1:* и *:*, для настройки последних двух вариантов используют формулы и скрипты мэппинга, реализованные в ETL-инструменте.

4. Процесс агрегации данных — этот процесс нужен из-за разности детализации данных в OLTP (система транзакций) и OLAP (аналитическая система) системах.

5. Выгрузка в целевую систему — это технический процесс использования коннектора и передачи данных в целевую систему.

Заключение

На сегодняшнем уроке мы с вами обсудили что же такое бизнес-аналитика и для чего она нужна, поговорили об анализе данных и его основных этапах, да и в целом для чего он собственно нужен. Также мы посмотрели, как можно применять python для задач связанных с анализом данных, а именно как производить обработку данных используя python. В последней части урока мы с вами поговорили о ETL

процессах, обсудили что же это такое и почему это важно и нужно для дата сайнтиста.

Спасибо за внимание и до новых встреч.