УДК 338

ОПИСАТЕЛЬНЫЙ И РАЗВЕДОЧНЫЙ АНАЛИЗ НА ЯЗЫКЕ PYTHON ДЛЯ ИССЛЕДОВАНИЯ БИЗНЕС-ДАННЫХ

Казаков Илья Александрович, студент бакалавриата,
Рындина Светлана Валентиновна, к.ф.-м.н., доцент
Пензенский государственный университет
Kazakov Ilya Alexandrovich, bachelor's degree student, kazakoff64ru@yandex.ru
Ryndina Svetlana Valentinovna, PhD in Physics and Mathematics, svetlanaR2004@yandex.ru
Penza State University

Аннотация. Бизнес-аналитика в компании строится на исследовании бизнес-данных: данных о поставщиках, потребителях, данных об уровне обслуживания, о сделанных заказах и проданных услугах/продуктах. Аналитика может быть реализована с использованием аналитических платформ и облачных сервисов, не требующих программирования, а может опираться на работу с кодом при выполнении различных видов анализа. В статье рассматривается бизнес-аналитика с использованием языка Python и готовых библиотек на этом языке.

Ключевые слова: анализ, данные, Python, визуализация, графики, диаграммы.

Описательный и разведочный анализ данных являются важными инструментами для понимания бизнес-процессов и принятия обоснованных решений. В современном мире огромное количество данных генерируется каждую секунду, и умение анализировать эти данные становится все более ценным навыком. Описательный анализ помогает понять основные характеристики данных, такие, например, как среднее значение, стандартное отклонение и распределение. Разведочный анализ позволяет находить скрытые закономерности и тенденции в данных, что может быть полезно для выявления новых возможностей или проблемных областей в бизнесе. Изучение и интерпретация информации, содержащейся в бизнес-данных, превращает их в ценный актив компании: извлекаются знания полезные в принятии решений, планировании или формировании новых продуктов/услуг.

Для проведения описательного и разведочного анализа бизнес-данных чаще используют специализированные решения, например, BI-системы, которые широко представлены на рынке программного обеспечения отечественными разработками, в том числе и облачными решениями (Yandex DataLens). Эти решения являются low-code (низкокодовыми, не требующими навыков программирования) и имеют дружелюбный интерфейс. При использовании компаниями этих систем необходимо предусмотреть финансирование не только процессов аналитики, но и покупку и продление лицензий на программное обеспечение (ПО) или оплату использования облачных сервисов, что является подчас очень существенным центром затрат. Дополнительно такой выбор отягощается рисками отказа в обслуживании и прекращении поддержки, если разработчик принимает такое решение по каким-либо причинам.

Также для целей бизнес-аналитики часто используются и языки программирования. Языки программирования Python и R как решения для анализа данных востребованы и в научной сфере и в

коммерческой. Но первенство у языка Python, так как это мощный и гибкий инструмент, при этом достаточно простой в изучении и использовании. Широкий спектр возможностей для работы с данными в Python реализован посредством библиотек, объединяющих вспомогательные функции, классы и т.п. для выполнения различных задач анализа данных. Базовые библиотеки для проведения аналитики с использованием Python: Pandas, NumPy, Matplotlib и Seaborn. Их инструменты позволяют проводить описательный и разведочный анализ для изучения бизнес-данных с целью извлечения из них практической пользы.

Библиотека NumPy [1] позволяет обрабатывать данные, представленные массивами и матрицами, и производить над ними различные математические операции, в ней также реализованы многие функции для статистического анализа.

Библиотека Pandas [2] основана на библиотеке NumPy и позволяет работать с одномерными данными типа Series и двумерными данными типа DataFrame. Последний вариант самый востребованный в анализе бизнес-данных, так как большинство данных из первичных источников: веб-приложений, информационных систем, облачных сервисов и т.п., которые использует бизнес, могут быть извлечены/экспортированы в формате csv, а с помощью метода read_csv() библиотеки Pandas содержимое таких файлов считывается в переменную в виде таблицы данных. Таблица данных – это двумерные данные типа DataFrame, каждый столбец таблицы обладает собственным типом данных, независящим от типов данных других столбцов набора. Формат csv – это экономичный формат хранения данных в виде плоских таблиц, в которых данные каждой строки относятся к некоторому наблюдению, а значения разных показателей в одном наблюдении отделяются друг от друга некоторым символом, чаще всего запятой (сокращение csv – Comma-Separated Values, т.е. значения, разделённые запятыми).

Библиотеки Matplotlib [3] и Seaborn [4] содержат реализацию методов визуализации данных. При этом, как библиотека Pandas основана на библиотеке NumPy и расширяет ее возможности по обработке табличных данных, наиболее частого варианта представления бизнес-данных, так и библиотека Seaborn основана на библиотеке Matplotlib и также расширяет возможности визуального представления именно табличных данных.

Работа с данными начинается с выявления в них потенциальных проблем: наличия пропусков, дубликатов, неверного определения типов, наличия выбросов, дисбаланса в представительности отдельных категорий данных. Количественные признаки имеют тип данных: int (целые) и float (действительные/вещественные числа с плавающей запятой). Категориальные признаки обычно представлены текстовыми метками и имеют тип object (текстовые, строковые или символьные данные). Особым типом является тип данных, определяющий дату и время – datetime, который может быть представлен и числом и меткой категории. Чаще всего проблема с распознаванием типа данных возникает при отнесении количественных данных и данных даты и времени к более общему типу object.

Для более глубокого понимания данных из исходного набора прибегают к преобразованию данных с помощью группировки, фильтрации, сортировки и агрегирования. Также можно осуществлять объединение и разделение таблиц, исключать/заполнять пропуски в данных, проводить преобразование типов данных. Преобразование типов данных не всегда связано с преодолением ошибочного распознавания типа для конкретного показателя. Например, из количественных данных можно с потерей части информации получить категориальные признаки, сопоставив интервалы изменения количественного признака с некоторой текстовой меткой: конкретные значения процентных ставок можно заменить на метки (высокая/средняя/низкая) в зависимости от принадлежности заменяемого значения соответствующему интервалу изменения процентных ставок.

Категориальные данные используются для построения срезов и выборок по значениям категорий. Но для категориальных данных важна примерно одинаковая представительность отдельных категорий. Например, в данных о заказах для категориального показателя способ оплаты значение категории «оплата при получении» может быть представлено в несколько раз меньшим числом наблюдений, чем значение «предоплата». Иногда это проблема сбора данных, в некоторых случаях такой дисбаланс возникает в силу естественных причин и реально отвечает контексту бизнеса.

Если же уникальных значений в категории слишком много, и их число сопоставимо с количеством наблюдений, то такие данные нужны только в процессе слияния данных и/или их обогаще-

ния, когда этот показатель используется как поле для объединения данных двух наборов и более. Для группировки и агрегирования такие показатели бесполезны.

Для категориальных данных также существует проблема с ошибочно внесенными значениями. Так как категориальные данные обычно имеют достаточно компактный домен уникальных значений, то корректным является выбор значения из списка при внесении таких данных в систему. Или проверка шаблона, если такие данные должны удовлетворять некоторым синтаксическим правилам. Для количественных данных также можно предусмотреть проверку граничных значений из допустимого интервала, это не исключит все ошибки, но позволит предотвратить наиболее существенные.

Описательный анализ позволяет сделать выводы о качестве данных на основе описательных статистик. Для количественных показателей – это среднее значение, мода, медиана, дисперсия и т.п., для категориальных показателей – число уникальных значений, наиболее представленная в данных категория. Можно провести и более подробный анализ: оценить распределение, выбросы и т.п.

Выбросы – это нехарактерные по значению данные для количественного показателя. Чаще всего их причина – ошибки регистрации, сбои в работе систем, но в некоторых случаях их появление вполне естественный процесс. Так клиент с огромным числом заказов может быть искусственно созданным сотрудниками компании для получения максимальной скидки. Если такое поведение сотрудников не нарушает правила компании, то необходимо исключить подобные наблюдения из рассмотрения. Если компания считает, что это злоупотребление служебным положением, то до сотрудников доводится информация о недопустимости подобных практик и настраиваются специальные методы проверки в системе оформления заказов. Внимание к деталям позволяет попутно выявить проблемы в сборе данных, а возможно и в реализации бизнес-процессов компании, и настроить функционирование сервисов и регистрацию данных корректным образом.

Разведочный анализ позволяет описание данных визуализировать, например, в виде столбчатых диаграмм для категориальных признаков или гистограмм для числовых показателей. Для визуализации данных в Python часто используются библиотеки Matplotlib и Seaborn. Они предоставляют широкий спектр возможностей для создания различных графиков и диаграмм, таких как гистограммы, графики распределения, «ящики с усами» и многих других. Визуализация данных помогает визуально представить структуру данных, выявить закономерности и обнаружить потенциальные аномалии.

Параллельное рассмотрение таблиц с описательными статистиками данных и визуализаций, графически описывающих отдельные показатели и/или их взаимосвязи с использованием дополнительных возможностей: цвет, размер, вид маркера для отображения точки наблюдения, помогают обнаружить неочевидные закономерности в распределении данных, выявить тренды, выбросы и сформировать предположения о причинно-следственных связях между переменными.

Так метод describe(), примененный к показателю charges (расходы), выдает информацию о числе наблюдений, среднем значении, стандартном отклонении, минимальном значении, квартилях и максимальном значении (рис. 1), а функция построения графика boxplot() (такой график называется «ящик с усами») из библиотеки Matplotlib визуализирует большинство рассчитанных характеристик этого показателя (рис. 2).

```
count     1338.000000
mean     13270.422265
std      12110.011237
min       1121.873900
25%       4740.287150
50%       9382.033000
75%      16639.912515
max      63770.428010
Name: charges, dtype: float64
```

Рисунок 1. – Пример описательного анализа для показателя расходы

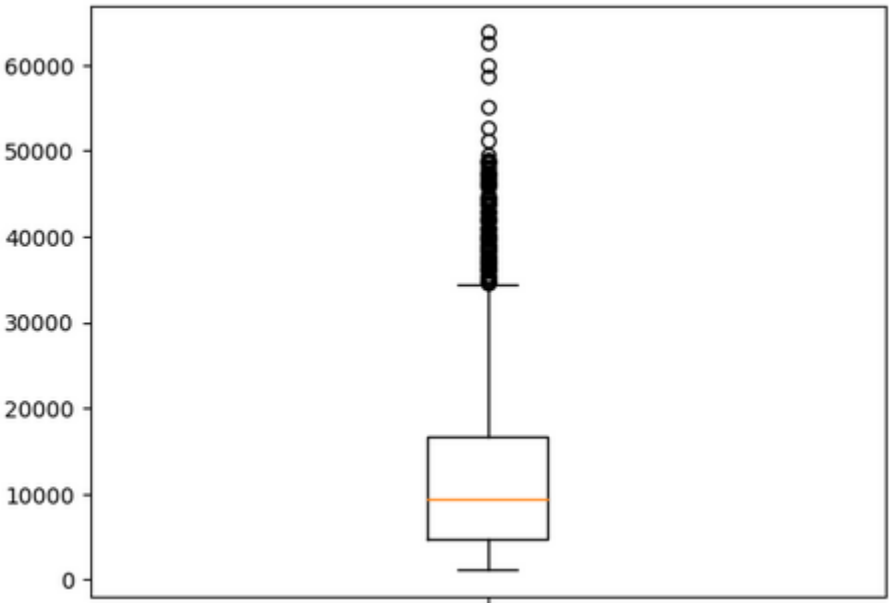

Рисунок 2. – Пример разведочного анализа для показателя расходы

Для компаний использование языка Python в аналитике бизнес-данных сопряжено как с преимуществами (независимость от проприетарных решений, экономия на обслуживании лицензионного ПО), так и с проблемами поиска и найма компетентного персонала, способного работать с данными таким образом.

Список использованных источников

1. NumPy. Официальный сайт. URL: https://numpy.org/ (дата обращения 09.04.2024)

2. Pandas. Официальный сайт. URL: https://pandas.pydata.org/ (дата обращения 09.04.2024)

3. Matplotlib: Visualization with Python. Официальный сайт. URL: https://matplotlib.org/ (дата обращения 09.04.2024)

4. Seaborn: statistical data visualization. Официальный сайт. URL: https://seaborn.pydata.org/ (дата обращения 09.04.2024)