|  |  |  |
| --- | --- | --- |
| *СИСТЕМЫ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ* | | |
| Лабораторная работа №1  «Разведочный анализ данных с помощью PySpark» | Группа | САПР-1.1 |
| Выполнил | Рублев А.А. |
| Проверил | Кравченя П.Д. |
| Дата выполнения | 29.09.2023 |
| Дата проверки |  |
| Подпись |  |

**Цель и задачи работы**

1. Познакомиться с понятием «большие данные» и способами их обработки;
2. Познакомиться с инструментом Apache Spark и возможностями, которые он предоставляет для обработки больших данных;
3. Получить навыки выполнения разведочного анализа данных использованием pyspark.

**Постановка задания**

Выполнить разведочный анализ датасета согласно варианту с определением:

* типов признаков в датасете;
* пропущенных значений и их устранением;
* выбросов и их устранением;
* расчетом статистических показателей признаков (средних, квартилей и т.д.);
* визуализацией распределения наиболее важных признаков;
* корреляций между признаками.

Вариант №2: "Датасет авиабилетов из Expedia".

Источник: https://www.kaggle.com/datasets/dilwong/flightprices

**Решение**

Датасет содержит информацию о ценах на полеты в одну сторону самолетами согласно Expedia на период с 16.04.2022 до 05.10.2022.

Датасет представляет собой CSV-файл, где каждая строка - купленный билет в/из следующих аэропортов: ATL, DFW, DEN, ORD, LAX, CLT, MIA, JFK, EWR, SFO, DTW, BOS, PHL, LGA, IAD, OAK.

Данные содержат следующую информацию:

* legId: Идентификатор рейса;
* searchDate: Дата (YYYY-MM-DD), когда эта запись была взята из Expedia;
* flightDate: Дата (YYYY-MM-DD) полета;
* startingAirport: Трехзначный код аэропорта IATA для начального местоположения;
* destinationAirport: Трехзначный код аэропорта IATA для места прибытия;
* fareBasisCode: [Код базового тарифа](https://en.wikipedia.org/wiki/Fare\_basis\_code);
* travelDuration: Продолжительность поездки в часах и минутах;
* elapsedDays: Количество прошедших дней (обычно 0);
* isBasicEconomy: Логическое значение, указывающее, предназначен ли билет для базовой экономики;
* isRefundable: Логическое значение, указывающее, подлежит ли билет возврату;
* isNonStop: Логическое значение, определяющее, является ли полет беспосадочным;
* baseFare: Цена билета (в долларах США);
* totalFare: Цена билета (в долларах США), включая налоги и другие сборы;
* seatsRemaining: Целое число для количества оставшихся мест;
* totalTravelDistance: Общее расстояние путешествия в милях, иногда эти данные отсутствуют;
* segmentsDepartureTimeEpochSeconds: Строка, содержащая время отправления (время Unix) для каждого этапа поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsDepartureTimeRaw: Строка, содержащая время отправления (формат ISO 8601: ГГГГ-ММ-ДДТчч:мм:сс.000±[чч]:00) для каждого этапа поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsArrivalTimeEpochSeconds: Строка, содержащая время прибытия (время Unix) для каждого этапа поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsArrivalTimeRaw: Строка, содержащая время прибытия (формат ISO 8601: ГГГГ-ММ-ДДТчч:мм:сс.000±[чч]:00) для каждого этапа поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsArrivalAirportCode: Строка, содержащая код аэропорта IATA для места прибытия для каждого этапа поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsDepartureAirportCode: Строка, содержащая код аэропорта IATA для места отправления для каждого этапа поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsAirlineName: Строка, содержащая название авиакомпании, обслуживающей каждый этап поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsAirlineCode: Строка, содержащая двухбуквенный код авиакомпании, обслуживающей каждый этап поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsEquipmentDescription: Строка, содержащая тип самолета, используемого на каждом этапе путешествия (например, «Airbus A321» или «Boeing 737-800»). Записи для каждой из ветвей разделяются знаком «||»;
* segmentsDurationInSeconds: Строка, содержащая продолжительность полета (в секундах) для каждого этапа поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsDistance: Строка, содержащая пройденное расстояние (в милях) для каждого этапа поездки. Записи для каждой из ветвей разделяются знаком «||»;
* segmentsCabinCode: Строка, содержащая салон для каждого этапа поездки (например, «пассажирский»). Записи для каждой из ветвей разделяются знаком «||».

Данные представляют собой одиночные значения следующих типов данных: integer, double, string, date, boolean. Распределение типов колонок представлено в таблице 1.

Таблица 1 – Распределение типов данных в колонках

|  |  |  |
| --- | --- | --- |
| № | Название колонки | Тип данных |
| 0 | legId | string |
| 1 | searchDate | date |
| 2 | flightDate | date |
| 3 | startingAirport | string |
| 4 | destinationAirport | string |
| 5 | fareBasisCode | string |
| 6 | travelDuration | string |
| 7 | elapsedDays | int |
| 8 | isBasicEconomy | boolean |
| 9 | isRefundable | boolean |
| 10 | isNonStop | boolean |
| 11 | baseFare | double |
| 12 | totalFare | double |
| 13 | seatsRemaining | int |
| 14 | totalTravelDistance | int |
| 15 | segmentsDepartureTimeEpochSeconds | string |
| 16 | segmentsDepartureTimeRaw | string |
| 17 | segmentsArrivalTimeEpochSeconds | string |
| 18 | segmentsArrivalTimeRaw | string |
| 19 | segmentsArrivalAirportCode | string |
| 20 | segmentsDepartureAirportCode | string |
| 21 | segmentsAirlineName | string |
| 22 | segmentsAirlineCode | string |
| 23 | segmentsEquipmentDescription | string |
| 24 | segmentsDurationInSeconds | string |
| 25 | segmentsDistance | string |
| 26 | segmentsCabinCode | string |

Всего в датасете 999998 строк.

После получения первого представления о колонках, в первую очередь необходимо убедиться, имеет ли датасет какие-либо пропущенные и аномальные нулевые значения.

* Для строковых колонок проверим на None и null;
* Для числовых колонок, допускающих значения ноль, проверим на None и NaN;
* Для числовых колонок, недопускающих значения ноль, проверим на нули, None и NaN;
* Для булевых колонок проверим на None и null;
* Для колонок с датами проверим на None и null.

Количество удаленных записей за каждый из типов колонок представлен в таблице 2.

Таблица 2 – Удаленное количество строк за каждую колонку

|  |  |  |
| --- | --- | --- |
| № | totalTravelDistance | segmentsEquipmentDescription |
| 0 | 61860 | 18754 |

Количество строк после удаления пропущенных и аномальных нулевых значений – 921949.

Перед обработкой данных сперва взглянем на статистические показатели числовых признаков, чтобы иметь представление о том, какие данные можно считать аномальными.

Рассчитаем такие показатели, как:

* Минимальное значение;
* Среднее значение;
* Максимальное значение;
* Среднеквадратичное отклонение.

Результаты поиска статистических показателей числовых признаков показан на рис. 1.

A screenshot of a computer code

Description automatically generated

Рисунок 1 – Статистические показатели числовых признаков

Прежде чем удалить выбросы, необходимо сначала решить, что считается выбросом. Есть два распространенных способа сделать это:

1. Использование межквартильного диапазона

Межквартильный размах (IQR) — это разница между 75-м процентилем (Q3) и 25-м процентилем (Q1) в наборе данных. Он измеряет разброс средних 50% значений.

Наблюдение можно определить как выброс, если оно в 1,5 раза превышает межквартильный размах, превышающий третий квартиль (Q3), или в 1,5 раза превышает межквартильный размах, меньше первого квартиля (Q1).

Выбросы = наблюдения > Q3 + 1,5\*IQR или Q1 – 1,5\*IQR.

2. Использование z-значения

Z-оценка показывает, сколько стандартных отклонений данного значения от среднего. Используем следующую формулу для расчета z-показателя:

z = (X - μ) / σ

где:

* X - это одно необработанное значение данных;
* μ - среднее значение населения;
* σ - стандартное отклонение населения.

Можно определить наблюдение как выброс, если его z-оценка меньше -3 или больше 3.

Выбросы = наблюдения с z-показателями > 3 или < -3.

Результаты работы методов Z-оценки и межквартильного диапазона представлены на рис. 2-3.

A screenshot of a computer program

Description automatically generated

Рисунок 2 – Результаты работы метода Z-оценки

A screenshot of a computer

Description automatically generated

Рисунок 3 – Результаты работы метода межквартильного диапазона

Как можно увидеть, метод межквартильного диапазона убирает довольно большую часть данных, так как граница в 75%, выше которой данные считаются аномальными, в случае, например, с elapsedDays, является довольно грубой. Конкретно с текущим датасетом использование данного метода не является наилучшим.

Метод Z-оценки является более оптимальным, поскольку обнаруживает действительно слишком отличающиеся значения.

Чтобы выявить связи между признаками, можно построить матрицу корреляций, преобразовав категориальные признаки в числовые. Матрица корреляций показана на рис. 4.

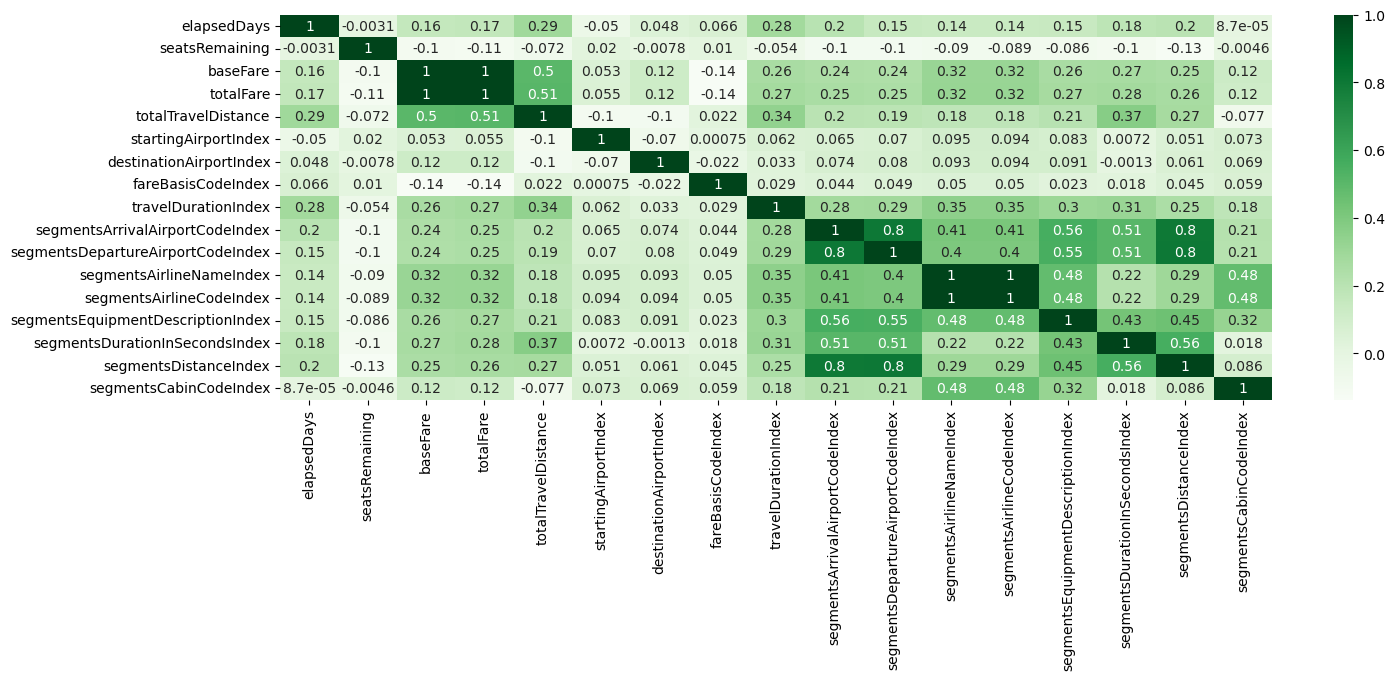


Рисунок 4 – Корреляционная матрица

Как видно из корреляционной матрицы:

* baseFare и totalFare имеют сильную положительную корреляцию, поскольку, очевидно, что на одну и ту же цену билета всегда будет одна и та же сумма налогов;
* baseFare, totalFare и totalTravelDistance имеют среднюю положительную корреляцию, так как цена билета в основном зависит от общего пройденного расстояния;
* segmentsAirlineNameIndex и segmentsAirlineCodeIndex имеют сильную положительную корреляцию, поскольку, очевидно, что название авиакомпании всегда будет имеет один и тот же код авиакомпании;
* segmentsArrivalAirportCodeIndex, segmentsDepartureAirportCodeIndex и segmentsDistanceIndex имеют сильную положительную корреляцию, так как расстояние между местом отправления и местом прибытия от рейса к рейсу примерно одно и то же;
* … и т.д.

**Выводы**

В ходе лабораторной работы был проведен разведочный анализ данных с помощью PySpark: определены типы признаков в датасете, устранены пропущенные значения и выбросы, рассчитаны статистические показатели признаков, визуализированы распределение наиболее важных признаков и корреляция между ними.