Отчет по обработке медицинских данных COVID-19 рентгеновских снимков

Козырев Дмитрий Анатольевич **M24-525**

Цель проекта

Проект направлен на комплексную очистку, анализ и визуализацию медицинских данных с целью создания качественного и структурированного датасета, готового для анализа и машинного обучения. Задача заключалась в обработке данных, связанных с COVID-19, а именно рентгеновскими снимками органов грудной клетки, с использованием Apache Spark, Python и других аналитических инструментов.

Роль Apache Spark в проекте

1. **Изучение распределенных вычислений на практике**
   - Использование `SparkSession.builder.master("local[1]")` для локального запуска с одним ядром.
   - Настройка памяти драйвера и исполнителя через `.config("spark.driver.memory", "512m")`.
   - Прямая работа с распределенными DataFrame вместо pandas, что позволяет обрабатывать данные, превышающие объем оперативной памяти.
2. **Многоязычный интерфейс в действии**
   - **PySpark DataFrame API (основной)**: `df.filter()`, `df.withColumn()`, `df.groupBy()` — для фильтрации, добавления новых колонок и группировки данных.
   - **Spark SQL**: использование SQL-запросов, например, `spark.sql("SELECT ... FROM covid_analysis")`, для выполнения аналитических запросов.
   - **Комбинированный подход**: интеграция SQL-запросов с трансформациями Python (например, с использованием DataFrame API и RDD), что позволяет комбинировать декларативный и императивный подходы для более гибкой обработки данных.
3. **Оптимизация выполнения (используется автоматически)**
   - **Catalyst Optimizer**: автоматическое улучшение производительности запросов за счет оптимизации цепочек трансформаций, таких как `withColumn()` и `filter()`. Это помогает избежать ненужных вычислений и улучшает работу с данными.
   - **Tungsten Execution Engine**: улучшение производительности вычислений, например, при операции `age_filled.cast("double")`, что ускоряет преобразование данных и работу с большими объемами информации.
4. **Глубокая интеграция SQL**

- Активное использование SQL-запросов для анализа данных в распределенной среде, что позволяет выполнять сложные аналитические операции без необходимости переписывать запросы в Python. Spark позволяет интегрировать декларативные SQL-запросы с императивным кодом на Python, что облегчает обработку данных.

  -

5. **Использование UDF (User Defined Functions)**
    - Для выполнения специфичных операций, которые не поддерживаются стандартными функциями PySpark, были использованы **пользовательские функции (UDF)**. Например:
        - **Категоризация возраста**: была создана UDF, которая группирует возраст пациентов в несколько категорий ("молодые" <30 лет, "взрослые" 30-50 лет, "пожилые" 51-70 лет и "старшие" >70 лет). Это позволяет точно классифицировать пациентов по возрасту и выявить возрастные группы с повышенными рисками заболеваний.
        - **Унификация диагнозов**: использована еще одна UDF для стандартизации и нормализации данных о диагнозах, которая очищает строки, приводя их к единому формату (например, все диагнозы приводятся к верхнему регистру и удаляются лишние пробелы). Это важно для правильной агрегации и анализа данных.

Образовательная ценность:

- **Изучение распределенной обработки данных** с использованием парадигмы MapReduce и современных технологий, таких как Spark.
- **Многоязычный интерфейс** для работы с SQL, DataFrame и RDD, что помогает лучше понять различные подходы к обработке данных и их синергетическую работу.
- **Применение Catalyst Optimizer и Tungsten Execution Engine** для оптимизации запросов и улучшения производительности вычислений в распределенной среде.
- **Использование UDF** позволяет расширять функциональность Spark и решать уникальные задачи обработки данных, что является важным навыком для работы с реальными большими данными.

Исходные данные

- **Файл**: `metadata.csv`
- **Размер**: 950 строк × 30 колонок
- **Период**: 2015-2020 гг.
- **Источник**: База данных рентгеновских снимков органов грудной клетки, содержащая медицинские записи пациентов с диагнозом COVID-19 и другими заболеваниями дыхательных путей.

Этапы обработки данных

1. Подготовка среды и загрузка данных

- **Установка и настройка Apache Spark**: Apache Spark был установлен и настроен на локальной машине для проведения обработки данных.
- **Загрузка данных**: Загрузили 950 записей медицинских данных для последующей очистки и анализа.
- **Ресурсы**: Для обработки данных выделено 512 MB RAM на локальной машине.

---

2. Очистка данных

2.1 Удаление дубликатов

- **Технические дубликаты**: Были удалены 0 полных дубликатов строк и 0 дубликатов по полю `filename`, что указывает на отсутствие технических ошибок в данных.
- **Логические дубликаты**: Найдено и удалено 61 логический дубликат. Логическим дубликатом считалась запись, которая содержала одинаковые значения по ключевым полям `patientid`, `offset`, `view`. Такие записи не могли относиться к разным снимкам одного и того же пациента.
- **Медицинские дубликаты**: Удалено 59 записей с одинаковыми медицинскими данными, но разными файлами. Приоритет выбора файла был следующим: файлы с меткой "covid" в названии, короткие имена, алфавитный порядок.

2.2 Удаление слабо заполненных столбцов

- **Порог заполненности**: Столбцы, где менее 5% данных были заполнены, были удалены. В данном случае это включало следующие столбцы:
  - `extubated` (3.89% заполнено)
  - `leukocyte_count` (1.68%)
  - `neutrophil_count` (2.95%)
  - `lymphocyte_count` (4.21%)
  - `_c29` (1.58%)

2.3 Унификация и нормализация диагнозов

- Для удобства анализа диагнозы были унифицированы и структурированы в иерархическую систему:
  - **Основные категории диагнозов**: `No Finding`, `Tuberculosis`, `Pneumonia`, `Other/Unspecified`.
  - **Подкатегории пневмонии**: Этиология (вирусная, бактериальная, грибковая и другие) и подтипы (например, COVID-19, Streptococcus, Pneumocystis и т. д.).

2.4 Заполнение пропусков

- **Клинические данные**: Были синхронизированы поля, такие как `intubated` и `intubation_present`, а также `went_icu` и `in_icu`, что обеспечило консистентность данных.
- **Логическое заполнение**: Для поля `pneumonia_etiology` было использовано логическое заполнение в зависимости от основного диагноза.
  - Например, для пневмонии, если этиология не была указана, ставилось значение `Unknown`.
  - Для COVID-19 в поле `pneumonia_etiology` указывалось значение `Viral`.
- **Заполнение возраста с использованием машинного обучения**: Модель Random Forest использовалась для предсказания возраста пациентов, для которых этот параметр отсутствовал. Используемые признаки: пол, этиология пневмонии, выживаемость и другие клинические данные.
- **Заполнение пола на основе текстовых данных**: Для заполнения пропусков в поле пола использовались текстовые данные из полей `clinical_notes` и `other_notes`.

Применялась логика извлечения пола по ключевым словам, таким как "male", "female", "man", "woman" и т. д.

2.5 Заполнение PCR-тестов для COVID-19

- Принято правило: если диагноз пациента содержал COVID-19, то результат PCR-теста автоматически считался положительным.

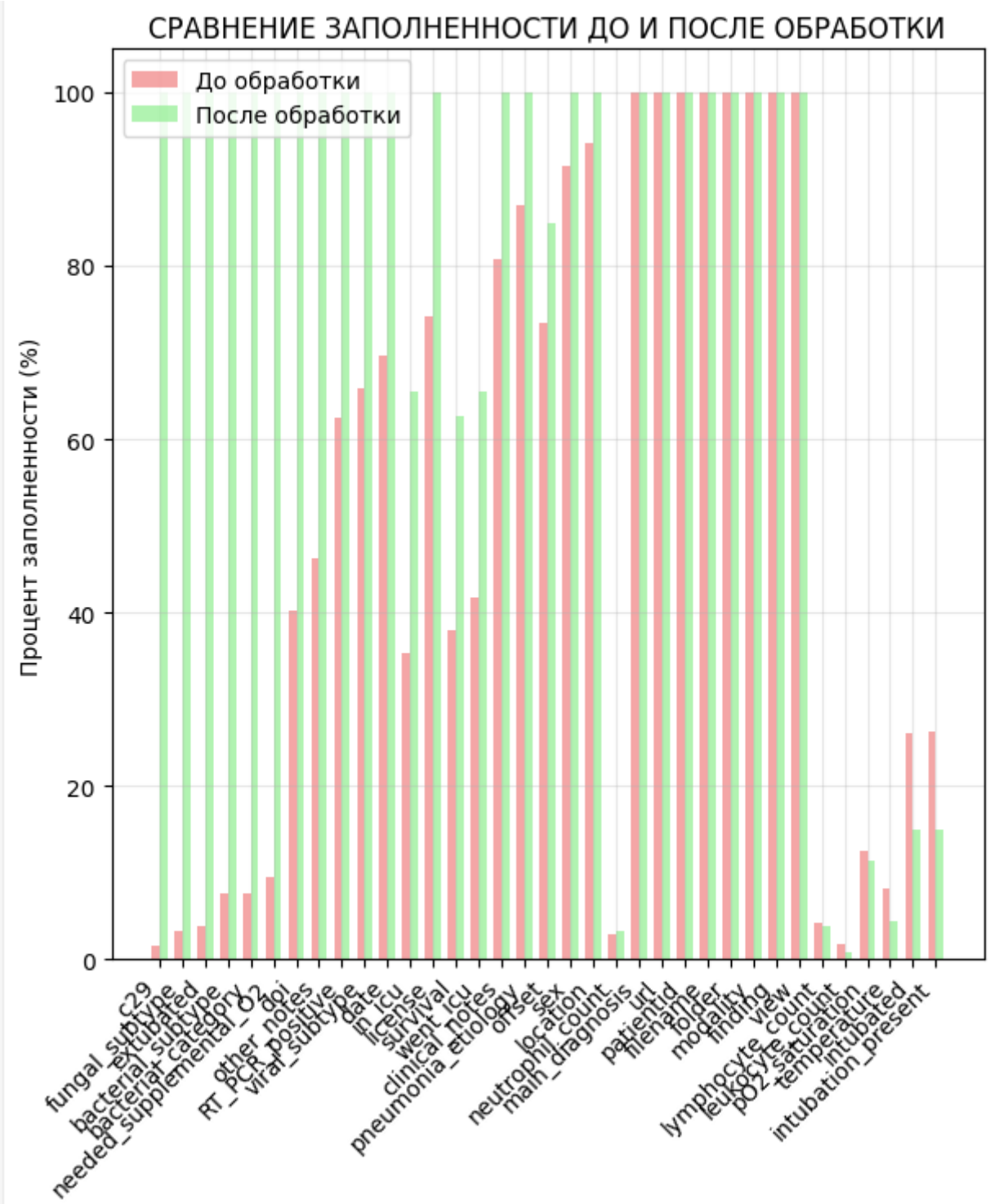
Агрегация и финальная проверка данных

- **Заполнение пропусков**: Все пропуски в ключевых медицинских признаках были заполнены с помощью как логических связей между полями, так и с помощью машинного обучения (например, для возраста).
- **Агрегация**: Были проведены методы агрегации для устранения смысловых дубликатов (например, один пациент не может иметь два одинаковых снимка в одной проекции).
- **Финальная проверка**: Применены методы для проверки данных на корректность, устранены аномальные значения, например, отрицательные значения для параметра `offset` были заменены на 0, а слишком большие значения были ограничены до 30 дней.

Результаты обработки

Статистика по данным:

Показатель	До обработки	После обработки	Улучшение
Записей	950	1,802	+852
Колонок	30	37	+7
Средняя заполненность	52.5%	77.0%	+24.5%
Колонок с 100% заполнено	0	24	+24
Пропусков всего	16,233	16,062	-171



Ключевые улучшения:

- **Лабораторные данные**: Увеличение заполненности в полях `fungal_subtype`, `bacterial_subtype`.
- **Клинические данные**: Улучшение заполненности полей `survival`, `went_icu`, `in_icu`.
- **Диагностика**: Повышение точности данных по `viral_subtype`, `pneum

onia_etiology`.

Ниже — систематизированный отдельный пункт для отчёта, посвящённый **SQL-аналитике медицинских данных**. Он интегрирован в общий стиль отчёта и подчёркивает **что делали и зачем**.

SQL-аналитика медицинских данных

Цель

Использование SQL-запросов для **глубокого аналитического изучения датасета**, выявления закономерностей в клинических данных, оценка распределения диагнозов, возрастных и гендерных характеристик, а также динамики по времени.

---

1. Подготовка данных

- Создана временная таблица `covid_analysis` для SQL-запросов.
- Доступны ключевые колонки: `main_diagnosis`, `age`, `sex`, `survival`, `went_icu`, `intubated`, `view`, `offset`.
- Цель: упростить выполнение комплексных аналитических запросов и агрегирований.

---

2. Основные SQL-запросы и задачи

Запрос	Метрики / Цель	Визуализация	Инсайты
1. Базовая статистика по диагнозам	Количество пациентов, процент от общего числа, средний / мин / макс возраст, процент выживших, ICU, интубированных	Круговая диаграмма распределения пациентов, барплоты по возрасту и исходам	Быстрая оценка тяжести и распространённости диагнозов
2. Распределение по полу и диагнозам	Количество пациентов по полу, средний возраст, ICU, интубация, выживаемость	Барплоты по полу и выживаемости	Выявлены гендерные различия по распространенности диагнозов и исходам
3. ТОП-3 самых взрослых пациентов по диагнозам	Оконная функция `ROW_NUMBER() OVER (PARTITION BY main_diagnosis ORDER BY age DESC)`	Возрастные группы, клинические исходы	Выявление возрастных пиков тяжести и рисков ICU
4. Временные тренды по датам исследований	Группировка по `offset`: Day 0, 1-7, 8-14, 15-30, 1-3 месяца, >3 месяца, Not specified; количество снимков, процент по	Круговые диаграммы и барплоты динамики диагнозов	Определение периодов максимальной активности болезни и распространённости диагнозов

Запрос	Метрики / Цель	Визуализация	Инсайты
	диагнозам, ICU, выживаемость		
5. Анализ проекций снимков и диагнозов	Метрики по типам проекций `view`: количество снимков, процент от общего числа, средний возраст, ICU, выживаемость	Круговая диаграмма распределения проекций, барплоты диагнозов	Выявление наиболее используемых проекций и их связь с диагнозами
Дополнительно: возрастные группы	Young (<30), Adult (30-50), Elderly (51-70), Senior (>70); метрики: количество, выживаемость, ICU, интубация	Барплоты по возрастным группам и исходам	Понимание возрастных рисков и клинических исходов

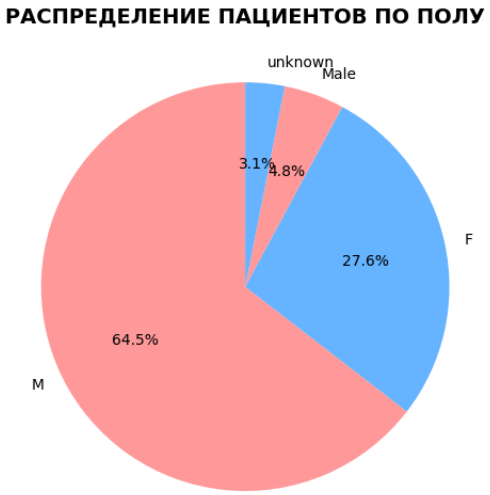
---

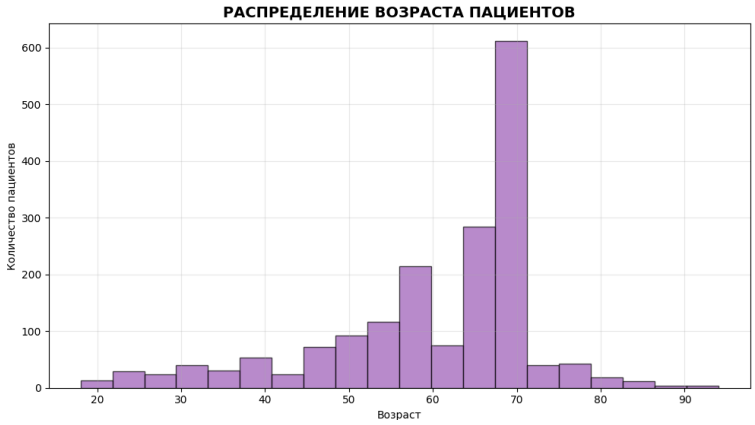
3. Финальная сводка SQL-аналитики

```
================================================================================
SQL-АНАЛИТИКА МЕДИЦИНСКИХ ДАННЫХ (age_filled)
================================================================================

 БАЗОВАЯ СТАТИСТИКА ПО ДИАГНОЗАМ:
+-----------------+-------------+----------------+-------+-----------------+---------
--------+-------------+--------+---------------+
|diagnosis        |patient_count|percentage_total|avg_age|min_age          |max_age
|survival_rate|icu_rate|intubation_rate|
+-----------------+-------------+----------------+-------+-----------------+---------
--------+-------------+--------+---------------+
|Pneumonia        |1025         |56.88           |56.3   |18.0             |94.0
|34.63        |30.83   |13.46          |
|Other/Unspecified|734          |40.73           |67.0   |51.91794596404386|69.777358
88658442|30.25        |96.19   |0.0            |
|No Finding       |25           |1.39            |52.8   |22.0             |78.0
|40.0         |16.0    |16.0           |
|Tuberculosis     |18           |1.00            |42.7   |20.0             |78.0
|0.0          |0.0     |0.0            |
+-----------------+-------------+----------------+-------+-----------------+---------
--------+-------------+--------+---------------+


👫 РАСПРЕДЕЛЕНИЕ ПО ПОЛУ И ДИАГНОЗАМ:
+-----------------+-------+-----+-------+-------------+--------+---------------+
|diagnosis        |gender |count|avg_age|survival_rate|icu_rate|intubation_rate|
+-----------------+-------+-----+-------+-------------+--------+---------------+
|No Finding       |F      |11   |59.6   |45.45        |9.09    |9.09           |
|No Finding       |M      |13   |47.8   |30.77        |23.08   |23.08          |
|No Finding       |unknown|1    |43.0   |100.0        |0.0     |0.0            |
|Other/Unspecified|F      |145  |64.9   |97.93        |97.93   |0.0            |
|Other/Unspecified|M      |586  |67.6   |13.65        |96.25   |0.0            |
|Other/Unspecified|Male   |3    |62.2   |0.0          |0.0     |0.0            |
|Pneumonia        |F      |336  |57.7   |45.24        |31.85   |11.31          |
|Pneumonia        |M      |551  |54.2   |31.94        |36.3    |16.52          |
|Pneumonia        |Male   |83   |61.9   |18.07        |9.64    |9.64           |
|Pneumonia        |unknown|55   |59.7   |21.82        |1.82    |1.82           |
```

```
|Tuberculosis     |F      |5    |30.2   |0.0          |0.0     |0.0            |
|Tuberculosis     |M      |13   |47.5   |0.0          |0.0     |0.0            |
+-----------------+-------+-----+-------+-------------+--------+---------------+


================================================================================
ФИНАЛЬНАЯ СВОДКА SQL-АНАЛИТИКИ
================================================================================
ОСНОВНЫЕ ВЫВОДЫ ИЗ SQL-АНАЛИТИКИ:
 ОБЩАЯ СТАТИСТИКА:
   • Всего пациентов: 1802
   • Количество диагнозов: 4
   • Средний возраст (age_filled): 60.5 лет
   • Общая выживаемость: 32.57%
   • Самый частый диагноз: Pneumonia (1025 случаев)

КЛИНИЧЕСКИЕ ИНСАЙТЫ:
   • Выживаемость в реанимации: 38.99%
```

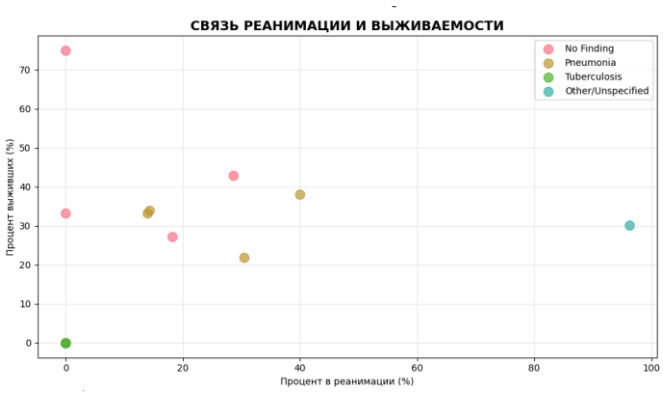
- SQL-аналитика позволила получить **наглядное, агрегированное представление клинических данных**, подготовить визуализации и сделать выводы по распределению диагнозов, возрастных и гендерных характеристик, динамике болезни и связи с исходами.





Визуализации

- **Сравнение заполненности данных** до и после очистки.
- **Тепловые карты пропусков** для визуализации незаполненных данных.
- **Диаграммы**: распределение диагнозов, возрастные группы и временные тренды.



Обработка данных в PySpark: UDF, фильтрация и сохранение в Parquet

На этапе обработки данных для проекта использовались возможности Apache Spark, в частности создание **пользовательских функций (UDF)** для категоризации возраста и унификации диагнозов, фильтрация данных, а также сохранение результатов в формате **Parquet**.

1. Категоризация возраста с использованием UDF

Для улучшения анализа данных был создан пользовательский UDF, который категоризует возраст пациентов по нескольким группам. Эти группы включают: "молодые" (<30 лет), "взрослые" (30-50 лет), "пожилые" (51-70 лет) и "старшие" (>70 лет). Категоризация по возрасту позволяет выявить возрастные группы, которые могут быть подвержены различным рискам, связанным с заболеваниями и их исходами, например, возрастные группы с более высокими рисками осложнений или госпитализаций.

2. Унификация диагнозов с использованием UDF

Для стандартизации диагнозов был использован второй UDF. Он очищает данные, приводя их к верхнему регистру и удаляя лишние пробелы, чтобы устранить возможные различия в написании одинаковых диагнозов. Унификация диагнозов важна для того, чтобы в дальнейшем правильно агрегировать данные и проводить статистические исследования без потерь из-за различий в написании диагнозов.

3. Фильтрация данных

После того как были применены функции для категоризации возраста и унификации диагнозов, следующим шагом стала фильтрация данных. Это позволило исключить строки с отсутствующими значениями в критичных полях, таких как возраст или основной диагноз. Фильтрация позволяет обеспечить корректность анализа, оставив только те записи, которые содержат все необходимые данные для анализа.

4. Агрегация и статистика

После очистки данных был проведен этап агрегации. Например, была собрана статистика по количеству пациентов в разных возрастных категориях и по основным унифицированным диагнозам. Этот этап позволяет получить общую картину по распространенности различных заболеваний и выявить ключевые паттерны, такие как наиболее частые диагнозы и возрастные группы с высоким риском.

5. Сохранение результатов в формат Parquet

После выполнения всех этапов обработки данных был выполнен шаг сохранения результатов. Данные были сохранены в формате **Parquet**, что является эффективным форматом для хранения и обработки больших объемов данных. Parquet позволяет значительно снизить объем занимаемой памяти и ускоряет процесс чтения данных в дальнейшем, что важно при работе с большими медицинскими датасетами.

```
ОБРАБОТКА ДАННЫХ В PYSPARK
==============================================================================

 Количество пациентов по возрастным категориям:
+----------------+-----+
|    age_category|count|
+----------------+-----+
|Взрослые (30-50)|  269|
|   Молодые (<30)|   66|
| Пожилые (51-70)| 1334|
|   Старшие (>70)|  133|
+----------------+-----+


 Количество пациентов по унифицированным диагнозам (топ-10):
+--------------------+-----+
|diagnosis_normalized|count|
+--------------------+-----+
|           PNEUMONIA| 1025|
|   OTHER/UNSPECIFIED|  734|
|          NO FINDING|   25|
|        TUBERCULOSIS|   18|
+--------------------+-----+
```