Проект: Аналитическая платформа для мониторинга COVID-19

Цель: Анализ рентгеновских снимков пациентов с диагнозами, включая COVID-19, пневмонию и другие заболевания. Используемые технологии: Hadoop, Spark, PySpark, ML (KMeans), SQL, визуализация (matplotlib).

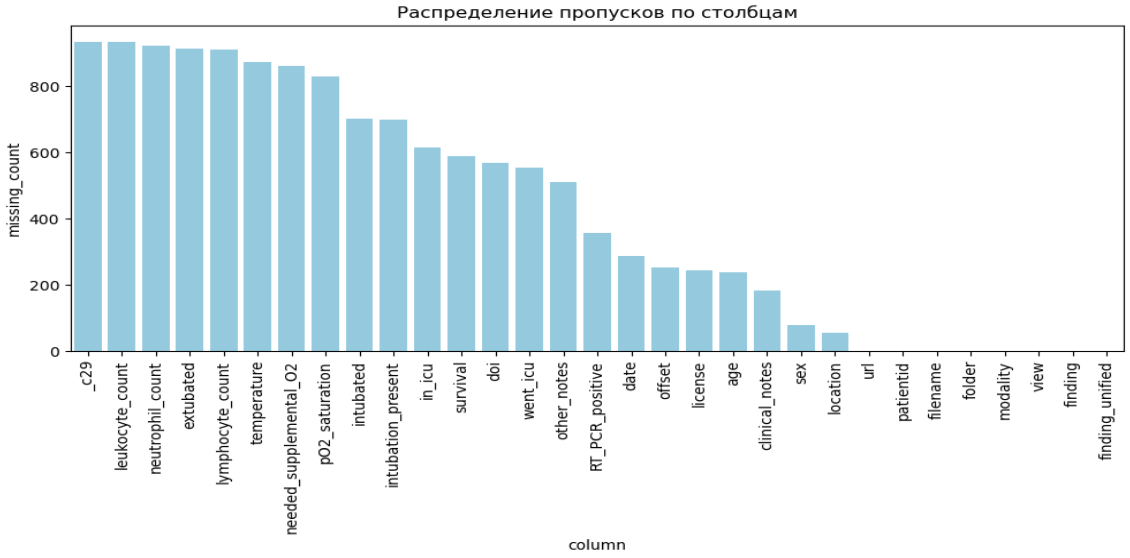
1. Схема и примеры оптимизаций

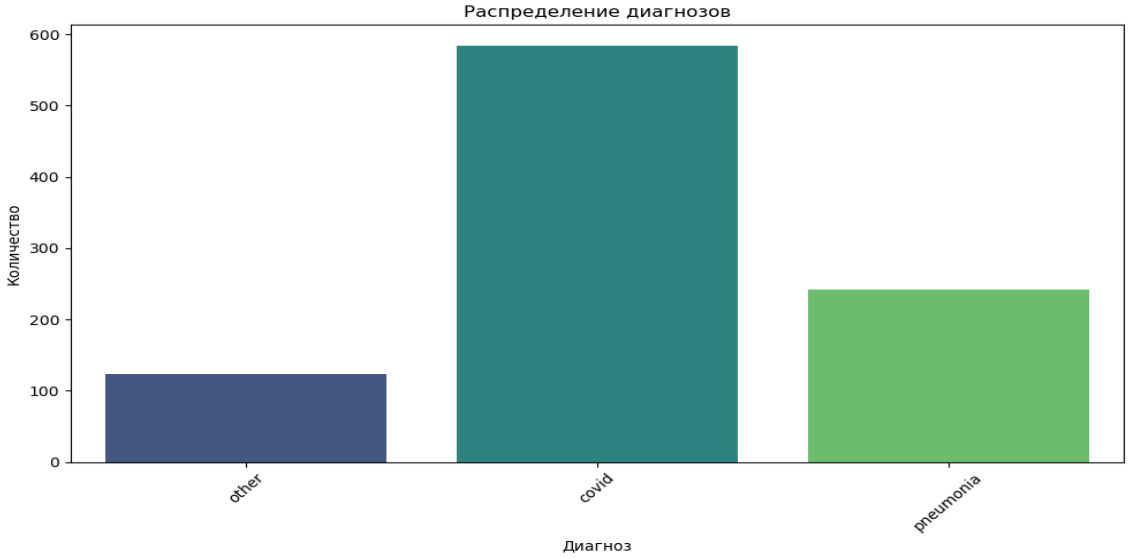
```
HDFS
    - metadata_cleaned.parquet
    - images/
        - covid/
        - pneumonia/
        -other/

Spark (ETL + ML)
    - Reading from HDFS
    - Cleaning and processing
    - KMeans Clustering
    - Saving in Hive

Hive (SQL-analytics)
    -covid_patients_optimized (partitioning by diagnosys)
    -patient_outcomes (bucketing by patientid)

Visualization
    -matplotlib (histograms, diagrams)
    -PDF-report (ReportLab)
```

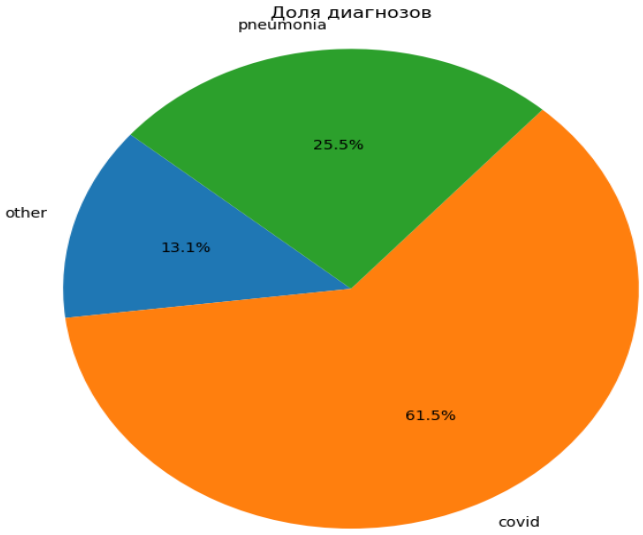
HDFS — хранение данных. Spark — обработка и ML. Hive — аналитика и хранение агрегированных данных. Визуализация — графики и PDF-отчёт.

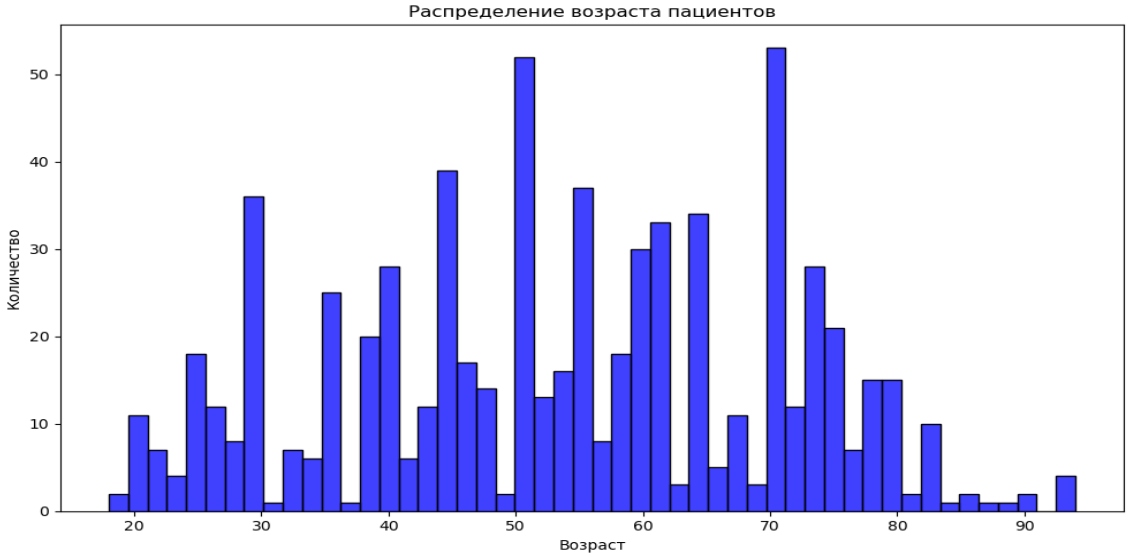
Партиционирование по диагнозам в Hive: ускоряет фильтрацию по finding_unified.
Бакетирование по patientid — ускоряет JOIN между таблицами. Сжатие Parquet (Snappy) — компромисс между скоростью и размером. Кэширование промежуточных результатов в Spark — ускоряет повторные вычисления.

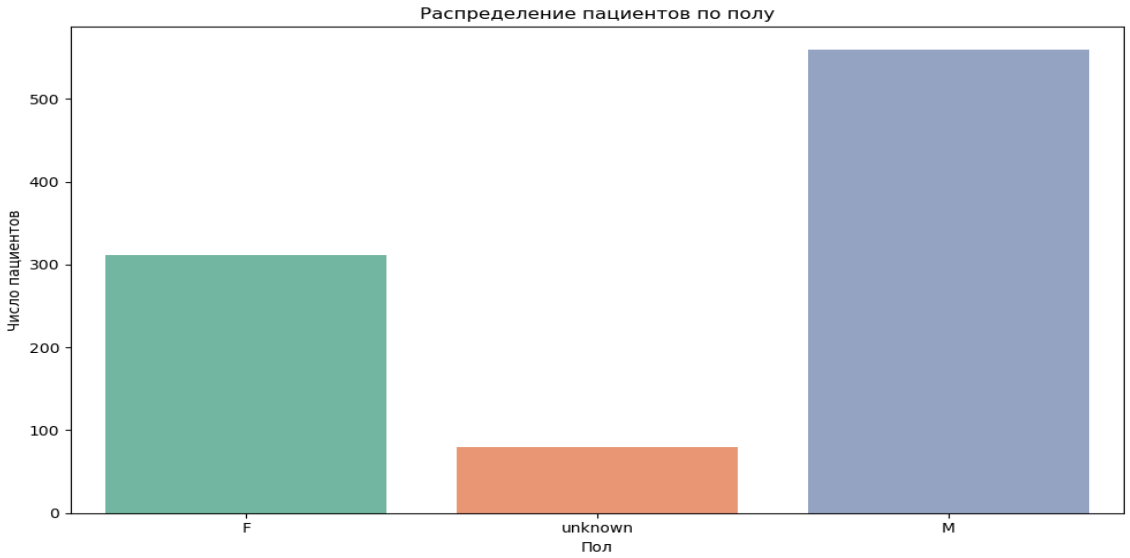
2. Визуализации

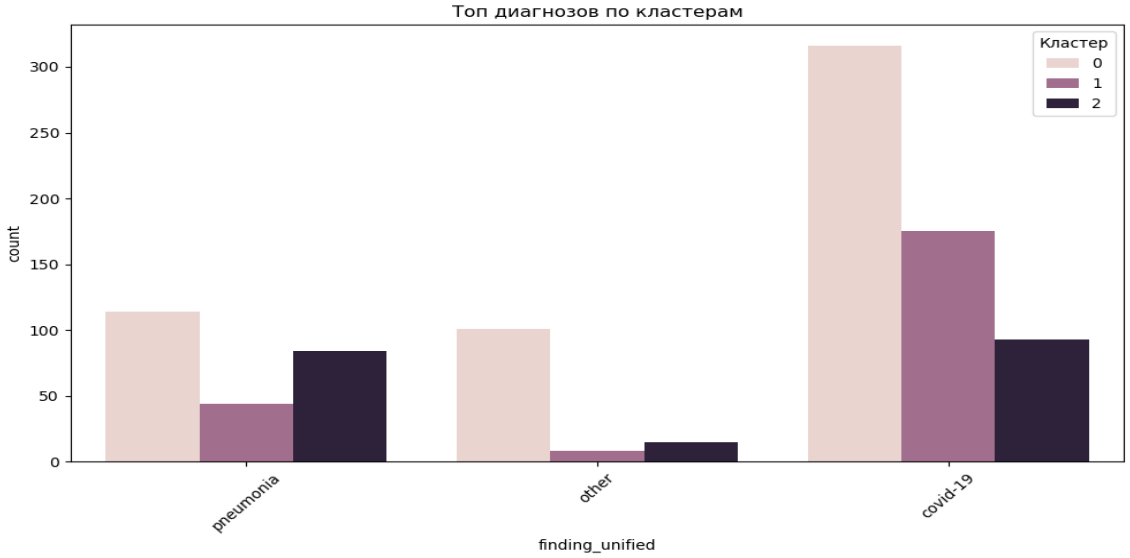

Распределение пропусков по столбцам
missing_count
0
200
400
600
800
_c29
leukocyte_count
neutrophil_count
extubated
lymphocyte_count
temperature
needed_supplemental_O2
pO2_saturation
intubated
intubation_present
in_icu
survival
doi
went_icu
other_notes
RT_PCR_positive
date
offset
license
age
clinical_notes
sex
location
url
patientid
filename
folder
modality
view
finding
finding_unified
column


Распределение диагнозов
Количество
0
100
200
300
400
500
600
other
covid
pneumonia
Диагноз


Доля диагнозов
pneumonia
25.5%
other
13.1%
61.5%
covid


Распределение возраста пациентов
Количество
0
10
20
30
40
50
20
30
40
50
60
70
80
90
Возраст





3. Анализ пропусков

Пропуски в данных были выявлены в полях `age`, `sex` и `clinical_notes`. Пропуски в числовых полях заполнены медианой. Категориальные пропуски заменены на "Unknown".

4. Интерпретация результатов

- 61.5% пациентов имеют диагноз COVID-19. - Кластер 0: пациенты среднего возраста (~52.7), в основном мужчины. - Кластер 1: пожилые пациенты (~73.0), равное распределение полов. -

Кластер 2: молодые пациенты (~31.7), в основном женщины.

5. Рекомендации по улучшению системы

1. Улучшение качества данных: Добавить автоматическую проверку качества новых данных при загрузке. Реализовать проверку на соответствие схеме. 2. Машинное обучение: Использовать CNN для анализа изображений. Предсказывать исход заболевания на основе клинических данных. 3. Интерфейс пользователя: Добавить Dash-панель для фильтрации и экспорта. Реализовать визуализацию по возрасту, полу и диагнозам. 4. Автоматизация: Настроить cron-задачи для ежедневной загрузки новых данных. Автоматически обновлять аналитические отчёты.

6. Структура HDFS

```
/covid_dataset/
    /images/
        /covid/
        /pneumonia/
        /other/
    /metadata/
        metadata_cleaned.csv
        metadata_cleaned.parquet
    /processed/
```

7. SQL-запрос с использованием оконной функции

```
print("1. Window function partitioned by diagnosys:")
spark.sql('''
SELECT
    patientid,
    finding_unified,
    age_cleaned,
    AVG(age_cleaned) OVER (
        PARTITION BY finding_unified
        ORDER BY patientid
        ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING
    ) AS moving_avg_age,
    COUNT(*) OVER (PARTITION BY finding_unified) AS patients_in_diagnosis
FROM covid_patients_optimized
WHERE finding_unified IS NOT NULL
LIMIT 20
''').show(truncate=False)
```

1. Оконная функция с партиционированием по диагнозу:

patientid	finding_unified	age_cleaned	moving_avg_age	patients_in_diagnosis
NULL	no_finding	75.0	68.0	22
38	no_finding	61.0	65.66666666666667	22
38	no_finding	61.0	64.0	22
173	no_finding	70.0	52.666666666666664	22
210	no_finding	27.0	45.333333333333336	22
211	no_finding	39.0	36.333333333333336	22
214	no_finding	43.0	34.666666666666664	22
217	no_finding	22.0	34.666666666666664	22
218	no_finding	39.0	33.333333333333336	22
218	no_finding	39.0	39.0	22
251	no_finding	NULL	39.0	22
253	no_finding	NULL	78.0	22
315	no_finding	78.0	66.0	22
316	no_finding	54.0	59.0	22
318	no_finding	45.0	48.0	22
318	no_finding	45.0	56.0	22
325	no_finding	78.0	62.666666666666664	22
351	no_finding	65.0	63.333333333333336	22
394	no_finding	47.0	55.666666666666664	22
397	no_finding	55.0	43.333333333333336	22

8. Заключение

Разработана аналитическая система на основе Hadoop, Spark и Hive. Выполнен анализ данных, выявлены кластеры пациентов. Создан отчет с визуализациями и рекомендациями. Дальнейшее развитие: внедрение ML-моделей, улучшение интерфейса, автоматизация. Обоснование принятых решений Использование HDFS и Spark : HDFS — масштабируемое хранение больших данных. Spark — позволяет обрабатывать данные в распределённом режиме, поддерживает ML и SQL. Партиционирование и бакетирование в Hive : Ускоряет выполнение запросов, особенно фильтрацию и JOIN. Снижает I/O при работе с большими наборами данных. Чистка и трансформация данных : Пропуски в числовых полях заполнены медианой. Пропуски в категориальных полях заменены на "Unknown". Диагнозы унифицированы для упрощения анализа. Кластеризация KMeans : Позволяет группировать пациентов по возрасту и полу. Может быть использовано для прогнозирования исхода заболевания.