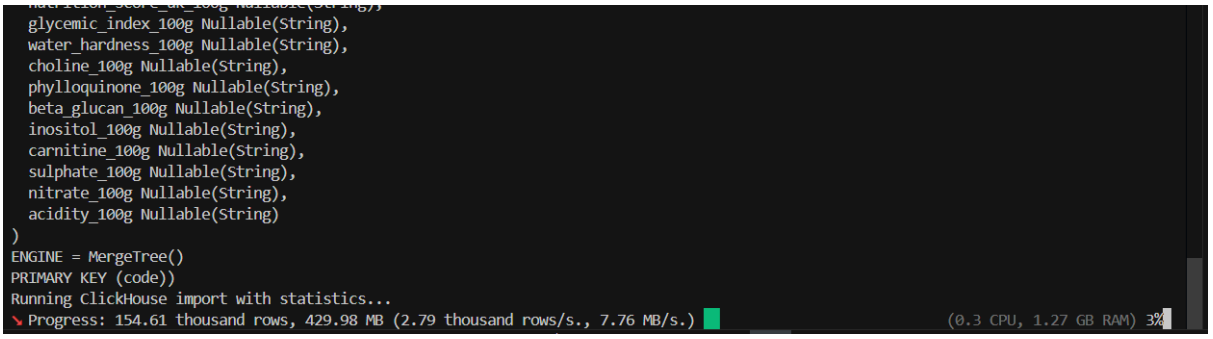
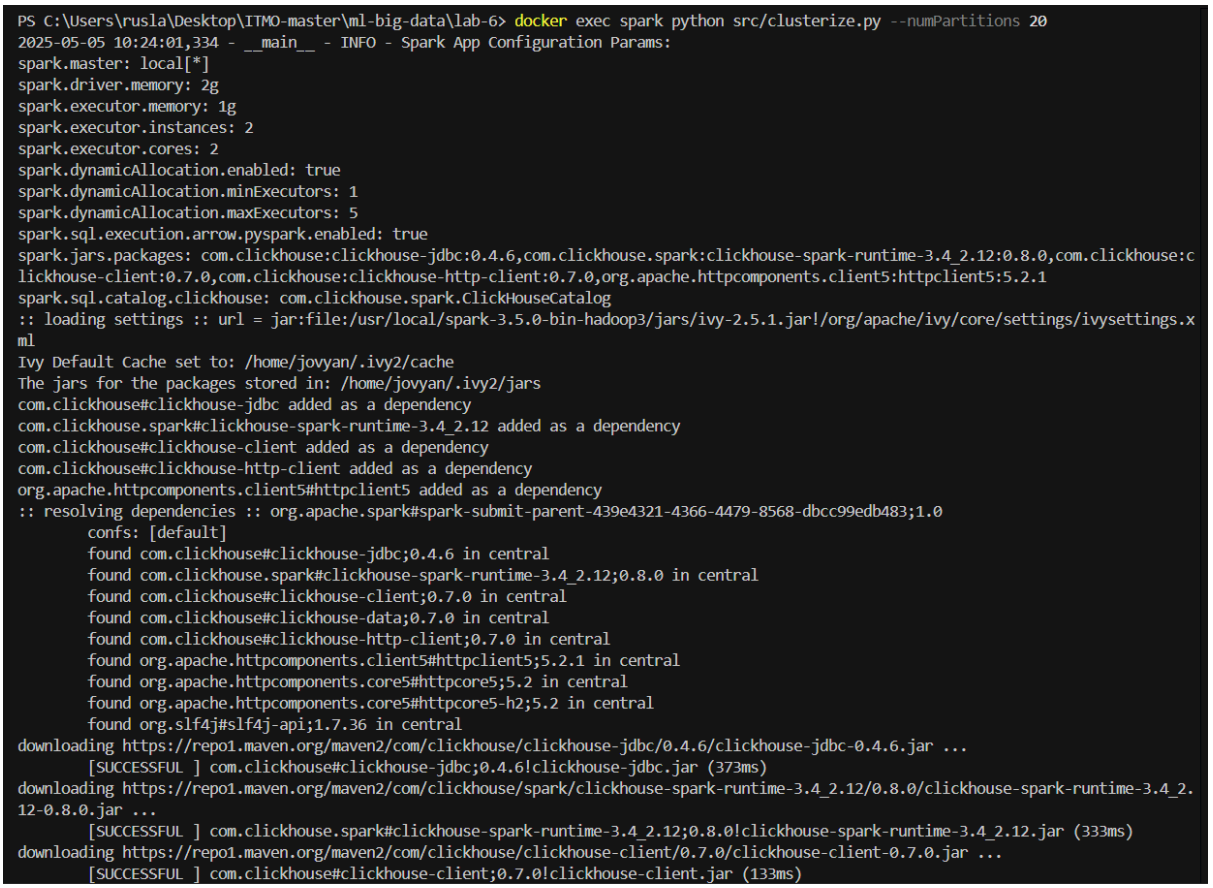
Отчет Лаб 6

Загрузка данных в базу ClickHouse



Запуск кластеризации и вывод параметров spark сессии



Логи работы скрипта кластеризации

```
        com.clickhouse#clickhouse-data;0.7.0 from central in [default]
        com.clickhouse#clickhouse-http-client;0.7.0 from central in [default]
        com.clickhouse#clickhouse-jdbc;0.4.6 from central in [default]
        com.clickhouse.spark#clickhouse-spark-runtime-3.4_2.12;0.8.0 from central in [default]
        org.apache.httpcomponents.client5#httpclient5;5.2.1 from central in [default]
        org.apache.httpcomponents.core5#httpcore5;5.2 from central in [default]
        org.apache.httpcomponents.core5#httpcore5-h2;5.2 from central in [default]
        org.slf4j#slf4j-api;1.7.36 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   9   |   9   |   9   |   0   ||   9   |   9   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-a5273e76-03ed-4529-97c3-b7f0efec9dda
        confs: [default]
        9 artifacts copied, 0 already retrieved (5299kB/99ms)
25/05/05 10:57:36 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where app
licable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/05/05 10:57:47 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. This behavior can be adj
usted by setting 'spark.sql.debug.maxToStringFields'.
2025-05-05 10:57:56,460 - __main__ - INFO - 6522 lines left after preprocessing
25/05/05 10:58:09 WARN InstanceBuilder: Failed to load implementation from:dev.ludovic.netlib.blas.VectorBLAS
2025-05-05 10:58:21,758 - __main__ - INFO - Class distribution: [1083, 2683, 917, 879, 960]
2025-05-05 10:58:27,453 - __main__ - INFO - Silhouette Score: 0.525312474839031 1
2025-05-05 10:58:37,350 - __main__ - INFO - Results saved successfully!
2025-05-05 10:58:37,363 - __main__ - INFO - Stopping spark app...
```

Первые 20 предсказаний из базы

```
f6c5acac7218 :) SELECT prediction FROM openfoodfacts LIMIT 20

SELECT prediction
FROM openfoodfacts
LIMIT 20

Query id: d69bee66-11ff-4e29-8c8c-387adc8487f3

    ┌─prediction─┐
 1. │       ᴺᵁᴸᴸ │
 2. │          0 │
 3. │          0 │
 4. │       ᴺᵁᴸᴸ │
 5. │       ᴺᵁᴸᴸ │
 6. │          3 │
 7. │       ᴺᵁᴸᴸ │
 8. │          3 │
 9. │          1 │
10. │          0 │
11. │          3 │
12. │       ᴺᵁᴸᴸ │
13. │          3 │
14. │          0 │
15. │          3 │
16. │          3 │
17. │          1 │
18. │       ᴺᵁᴸᴸ │
19. │       ᴺᵁᴸᴸ │
20. │          3 │
    └────────────┘
```

Файл конфигурации spark

```
[spark]
spark.master = local[*]
spark.driver.memory = 2g
spark.executor.memory = 1g
spark.executor.instances = 2
spark.executor.cores = 2
spark.dynamicAllocation.enabled = true
spark.dynamicAllocation.minExecutors = 1
spark.dynamicAllocation.maxExecutors = 5
spark.sql.execution.arrow.pyspark.enabled = true
spark.jars.packages = com.clickhouse:clickhouse-jdbc:0.4.6,com.clickhouse.spark:clickhouse-spark-runtime-3.4_2.12:0.8.0,com.clickhouse:clickhouse-c
spark.sql.catalog.clickhouse = com.clickhouse.spark.ClickHouseCatalog

[model]
seed = 42
k = 5
```

Инициализация spark сессии

```
self.spark = SparkSession.builder.config(conf=conf) \
    .config("spark.sql.catalog.clickhouse.host", IP_ADDRESS) \
    .config("spark.sql.catalog.clickhouse.protocol", PROTOCOL) \
    .config("spark.sql.catalog.clickhouse.http_port", PORT) \
    .config("spark.sql.catalog.clickhouse.user", self.USER) \
    .config("spark.sql.catalog.clickhouse.password", self.PASSWORD) \
    .config("spark.sql.catalog.clickhouse.database", self.DATABASE) \
    .getOrCreate()
```

Чтение таблицы из базы при помощи jdbc с возможностью указать количество партиций

```
self.url = f"jdbc:clickhouse://{IP_ADDRESS}:{PORT}/{self.DATABASE}?socket_timeout={socket_timeout}"
self.driver = "com.clickhouse.jdbc.ClickHouseDriver"
self.index_col_name = "numeric_index"
self.query = f"""(SELECT *, rowNumberInBlock() AS {self.index_col_name} FROM {self.TABLE}) AS subquery"""
```

```
def run(self):
    df = self.spark.read.format('jdbc') \
        .option('driver', self.driver) \
        .option('url', self.url) \
        .option('dbtable', self.query) \
        .option('user', self.USER) \
        .option('password', self.PASSWORD) \
        .option("partitionColumn", self.index_col_name) \
        .option("lowerBound", "1") \
        .option("upperBound", "100") \
        .option("numPartitions", self.numParitions) \
        .option("fetchsize","10000") \
        .load()

    df.cache()
```

Сохранение результатов в базу

```
def save_results(self, init_df, transformed):
    self.spark.sql(f'TRUNCATE TABLE clickhouse.{self.DATABASE}.{self.TABLE}')

    init_df.drop(self.index_col_name, 'prediction') \
        .join(transformed.select(*self.metadata_cols, 'prediction'), on='code', how='left') \
        .write.mode("append") \
        .format("jdbc") \
        .option("driver", self.driver) \
        .option("url", self.url) \
        .option("dbtable", self.TABLE) \
        .option("user", self.USER) \
        .option("password", self.PASSWORD) \
        .option("batchsize","10000") \
        .save()
```

Чистим изначальную базу и на ее место записываем join основной таблицы и таблицы предсказаний по колонке ‘code’.