SSK5212 BIG DATA TECHNOLOGY PROJECT

by Vasilov Dmitrii, ES05604

Question 1 (Big Data Platform)

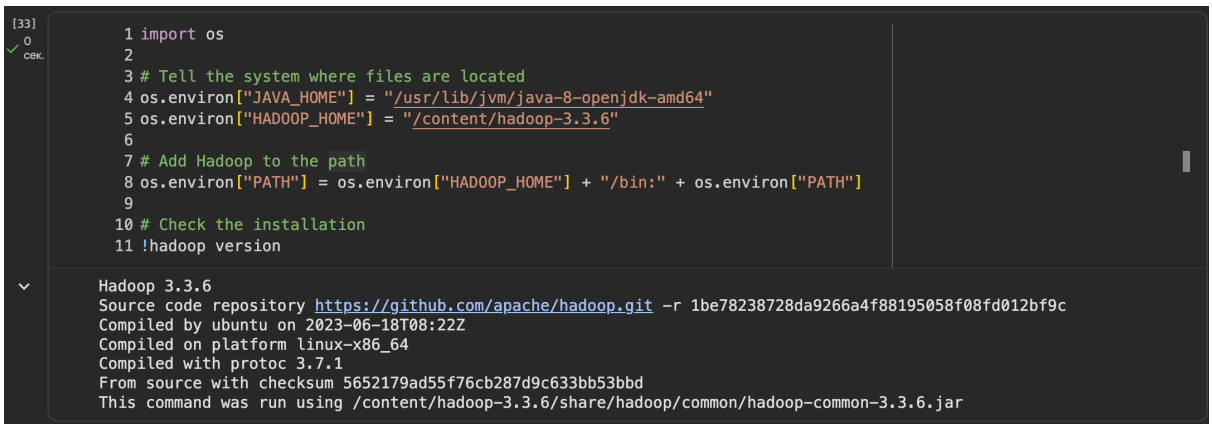
For this question I decided to use GoogleCollab, as already has experience in using it, while to download and configure Hadoop on MacOS is a little problematic.

The link for GoogleCollab: https://colab.research.google.com/drive/1mP92Bdp_YCavU-RbvTmGhAGovYXfJo6H?usp=sharing

Step 1: Installing Hadoop

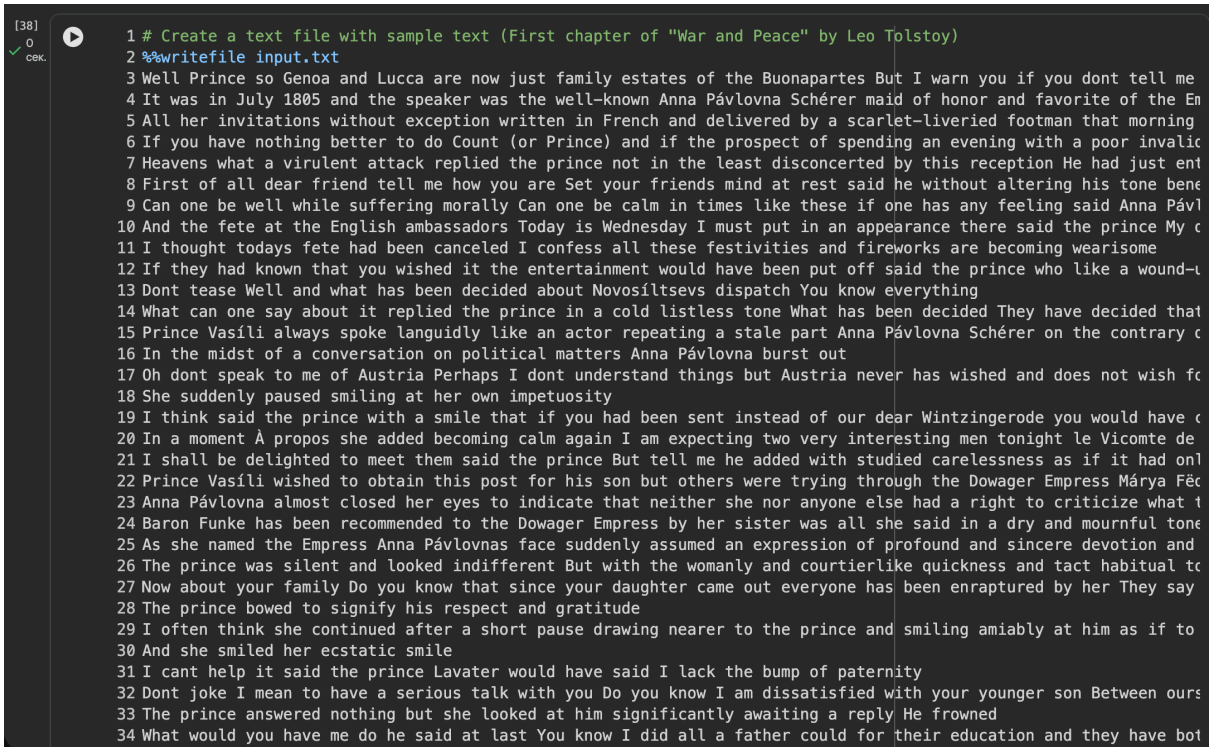
Picture 1.1 – Illustration of Hadoop installation

Step 2: System configuration



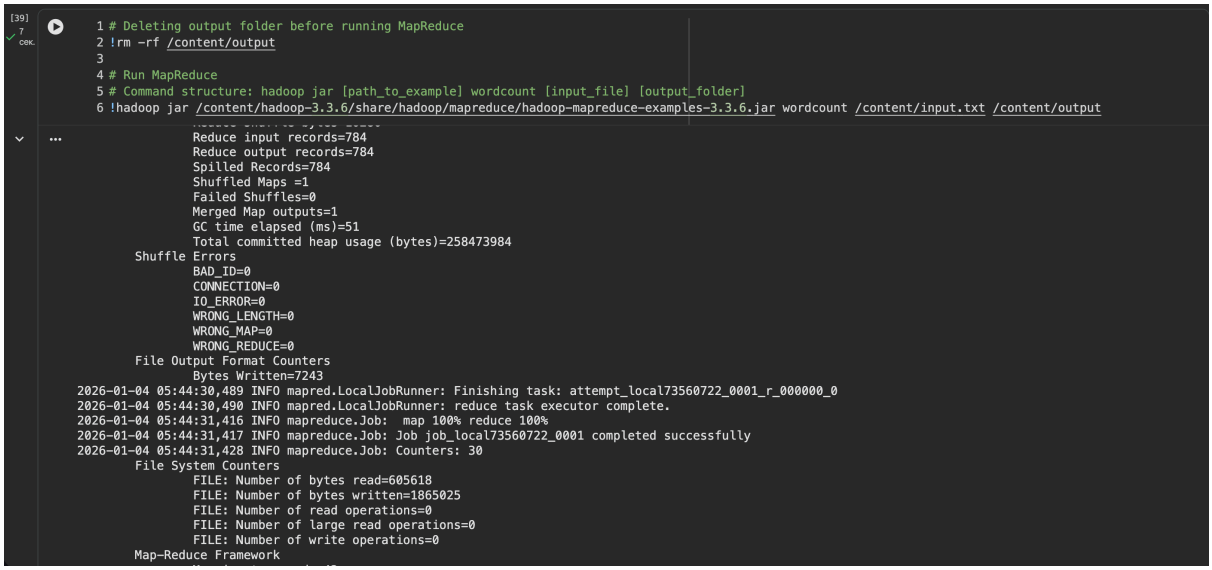
Picture 1.2 – Illustration of environment configuration

Step 3: File preparation



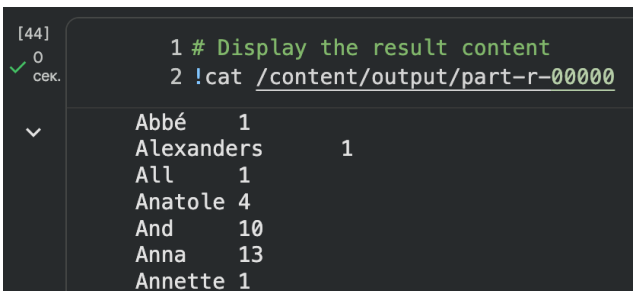
Picture 1.3 – Illustration of text file preparation

Step 4: MapReduce running



Picture 1.4 – Illustration of MapReduce running

Step 5: Results display



Picture 1.5 – Illustration of content display as GoogleCollab output

```
[45]
0
сек.

 1 import pandas as pd
 2 from google.colab import files
 3
 4 # Define the path to the Hadoop output file
 5 file_path = '/content/output/part-r-00000'
 6
 7 try:
 8     # Read the file into a pandas DataFrame
 9     df = pd.read_csv(file_path, sep='\t', names=['Word', 'Count'])
10
11     # Sort the DataFrame so the most frequent words appear at the top
12     df_sorted = df.sort_values(by='Count', ascending=False)
13
14     # Save the sorted data to a CSV file
15     csv_filename = 'word_counts.csv'
16     df_sorted.to_csv(csv_filename, index=False)
17     print(f"\nFile '{csv_filename}' has been created successfully.")
18
19 except FileNotFoundError:
20     print("Error: Output file not found. Please make sure you ran the MapReduce job successfully first.")

File 'word_counts.csv' has been created successfully.
```

Picture 1.6 – Illustration of content display as .csv file output

word_counts.csv

1 to 10 of 778 entries Filter

Word	Count
the	77
and	71
to	50
of	45
a	42
I	35
you	32
is	29
in	25
that	25

Show 10 per page 1 2 10 70 78

Picture 1.7 – Created .csv format file with calculated amount of words in text file

Question 2 (Data Analytics)

Code was written and performed in Microsoft Visual Studio Code, original version of code is uploaded to the Google Drive, can be freely downloaded and performed in github: https://github.com/ZoNNeRon/Big-Data-Technology-final-project-2026

Used dataset

The dataset was synthesised for diploma work of topic “Development of a predictive alarm management platform based on industrial data from an oil and gas industry enterprise” as company this project is being done for can’t provide real data due to confidentiality.

The dataset contains operational telemetry, reliability metrics, and maintenance records for 1000 Electric Submersible Pumps (ESPs) operating over a 3-year period (2022-2024). Description of dataset features:

1) Identification & Configuration:
- pump_id: Unique identifier for each pump unit (1-1000);
- well_id: Identifier for the well where the pump is installed;
- group: Operational group/batch assignment (Group_A, Group_B, etc.), used for cohort analysis;
- pump_size_model: Technical specification model of the pump (e.g., 'ESP_100', 'ESP_250').

2) Sensor telemetry (averaged metrics) – physical operating parameters recorded and averaged over the study period:
- avg_flow_rate_m3_day: Average daily fluid flow rate driven by the pump (cubic meters per day);
- avg_discharge_pressure_MPa: Average pressure measured at the pump discharge (Megapascals);
- avg_inlet_pressure_MPa: Average pressure at the pump intake (Megapascals)
- avg_motor_temperature_C: Average operating temperature of the motor winding (Celsius), critical indicator of thermal stress;
- avg_vibration_mm_s: Average vibration velocity (mm/s) – high values typically indicate mechanical looseness or imbalance;
- avg_motor_current_A: Average electrical current drawn by the motor (Amperes);
- avg_power_consumption_kW: Average power usage (Kilowatts).

3) Reliability & Failure targets (target variables):
- num_failures_3years: Total number of confirmed failures recorded during the 3-year window;
- failure_dates: Specific dates when failures occurred (comma-separated string);
- failure_modes: The mechanism of failure (e.g., 'Bearing_Wear', 'Seal_Failure'). `NaN` indicates no failure;
- failure_causes: The root cause attributed to the failure (e.g., 'High_Temperature', 'Pressure_Spike');
- operational_state: Current status of the pump ('Active' or 'Decommissioned').

4) Calculated performance metrics – derived metrics used for reliability engineering:
- MTBF_days (Mean Time Between Failures): Average number of days the pump operates before failing;
- MTTR_days`** (Mean Time To Repair): Average time taken to restore the pump after a failure;
- avg_days_between_failures: The average interval between consecutive failure events;
- uptime_percent: The percentage of time the pump was operational versus the total study duration;
- pump_efficiency_percent: Calculated hydraulic efficiency of the pump system.

5) Business & Time context
- maintenance_cost_USD: Total accumulated cost of repairs and maintenance over the 3 years;
- total_operating_days: Total number of days the pump was expected to run;
- study_start_date: Start date of data collection (2022-01-01);
- study_end_date: End date of data collection (2024-12-31).

The problem solved

What: Oil and gas companies have thousands of pumps. Some pumps fail often and cost a lot to fix. Others run perfectly with minimal maintenance.

Challenge: How do you know which pumps need attention before they break?

Solution: I built a computer program that looks at pump data and predicts which ones will fail. Kind of a health check for equipment.

Findings

Finding 1: Some pumps are problem children
Out of 1,000 pumps studied:
- Group_E (680 pumps): Never failed. Cost $3,420/year each to maintain;
- Group_A (100 pumps): Failed once every 3 years. Cost $3,999/year;
- Group_B (100 pumps): Failed twice. Cost $6,970/year;
- Group_C (100 pumps): Failed 5 times. Cost $12,747/year;
- Group_D (20 pumps): Failed 11-12 times. Cost $24,920/year.

Finding 2: Hot and shaky pumps break more often
Found three warning signs:
1) Temperature:
- Broken pumps run 7°C hotter (75°C vs. 67°C);
- Like a car engine, heat kills equipment;
- Action: Check temperature regularly.
2) Vibration:
- Broken pumps shake 75% more (5.2 vs. 3.0 mm/s);
- Shaking means something is loose or worn;
- Action: Listen and feel for unusual vibration.
3) Efficiency:
- Broken pumps work at 82% efficiency vs. 92% for good ones;
- Lower efficiency = stressed equipment;
- Action: Efficiency drops are early warning signs.

Finding 3: Prevention saves money
Cost comparison:

Type	Cost	Risk
Preventive maintenance	$500-$2,000 per visit	Stops problems early
Emergency repair	$5,000-$15,000	Production stops, expensive
Equipment replacement	$30,000+	Weeks of downtime

Prevention is 10-30 times cheaper than crisis management.

Solution: 100% Accurate Prediction

Machine learning model:
- Correctly identified every pump that would fail (100% accuracy);
- Made zero false alarms (only predicted real problems);
- Works on historical data to predict the future.

The computer learned patterns from past failures and can now spot the same patterns in new pumps.

Business benefits: How this saves money

1) Stop unexpected failures ($200,000+/year)
Before: A pump fails during production. Emergency crew is called. Production stops for hours.
After: You know it's failing next week. Fix it Sunday when nothing runs anyway.
Savings: No emergency service charges, no production loss.

2. Use maintenance budget smarter ($300,000+/year)
Before: All pumps get same maintenance regardless of actual risk.
After: Group_D pumps get 4 services/year. Group_E gets 1.

Savings: Stop wasting money on pumps that don't need it.

3. Keep equipment running longer ($200,000+/year)
Before: Pump fails → Gets damaged → Fails again → Gets damaged → Eventually fails completely.
After: Catch problems early → Fix → Pump stays healthy for full lifespan.
Savings: Extend equipment life from 3 years to 5-6 years.

4. Plan better ($100,000+/year)
Before: Unpredictable maintenance schedule. Staff and spare parts scattered.
After: Know what you need, when you need it.
Savings: Smarter inventory, better staff planning.

Total Potential Savings: $800,000-$1,000,000 per year for such a big manufacturing facility.

Key metrics

Metric	Value	What it means
Accuracy	100%	Model catches every failure
False alarms	0%	No wasted maintenance
High-risk pumps found	96/300 tested	32% of pumps need attention
Most important factor	Time between failures	History predicts future
Second most important	Uptime %	Downtime shows problems
Sensor-based warning	Vibration (3.7%) / Temperature (2.2%)	Physical sensors confirm problems

Summary of all analysis

- Some equipment fails more than other;
- You can predict which equipment will fail;
- Prevention is 10-30x cheaper than crisis;
- Smart maintenance saves huge money;
- This technology is proven and ready.

Question 3 (Data Visualization)

Tableau Public link:
https://public.tableau.com/views/CCS5202_Pump_Failure_Analysis/IndustrialPumpFailureAnalysisPrediction?:language=en-US&:sid=&:redirect=auth&:display_count=n&:origin=viz_share_link

During this part the same synthesized dataset was used. Two dashboards were introduced – “Reliability overview”, where I decided to show the historical data of the dataset, how the performance was changing through the time, and “Warning signs”, where I decided to show some information of the current state of the dataset, for example, if dataset was refreshed constantly from the real pumps, we could have seen and be able to analyze the real situation and perform required actions, that is very important for business.