

Report

AWS

Part 1 – Data Collection & Preprocessing

Goal:

Collect and prepare customer reviews for sentiment analysis.

Tools Used:

- **Python, BeautifulSoup, Requests, Pandas.**
- **deep_translator** for translation from Russian to English.
- **Amazon S3** for storage.

Steps Taken:

1. **Data Collection:** Scraped product titles and reviews from **Technodom.kz**.
2. **Translation:** Translated reviews using **deep_translator**.
3. **Data Cleaning:** Removed empty values, fixed encoding issues.
4. **Data Upload:** Uploaded cleaned data to **S3**.

Outcome:

Cleaned dataset uploaded to **S3** for further processing.

Part 2 – Data Preparation with AWS Glue

Goal:

Prepare the dataset for querying and transformation using **AWS Glue** and **Athena**.

Steps Taken:

1. **Glue Crawler:** Ran a **Glue crawler** to catalog the dataset.

Athena Fix: Fixed column headers by creating a new table with proper names:

```
CREATE TABLE my_data_catalog.technodom_reviews_fixed AS  
  
SELECT col0 AS original, col1 AS translated  
  
FROM my_data_catalog.technodom_reviews  
  
WHERE col0 != 'original';
```

2. **ETL Job:** Created an **ETL job** using **PySpark** to clean and deduplicate the data. Saved it to **S3**.

Outcome:

Dataset cleaned and validated, stored in **S3**, ready for analysis.

Part 3 – Sentiment Analysis Using Transformers

Goal:

Perform sentiment analysis using **HuggingFace Transformers**.

Steps Taken:

Model Setup: Used **HuggingFace sentiment-analysis pipeline**:

```
sentiment_analyzer = pipeline("sentiment-analysis")
```

Processing: Applied the model to the **translated** column:

```
df["label"] = df["translated"].apply(lambda text:  
sentiment_analyzer(str(text)[:512])[0]["label"].lower())
```

1. **Exporting Results:** Saved results to **sentiment_labels.csv** and uploaded to **S3**.

Outcome:

Sentiment-labeled data uploaded to **S3**, ready for visualization.

Part 4 – Visualization in Amazon QuickSight

Goal:

Visualize sentiment analysis results using **Amazon QuickSight**.

Steps Taken:

1. **Data Source:** Used **sentiment_labels.csv** for visualization.
2. **Manifest File:** Created **manifest.json** to define file location.
3. **QuickSight Connection:** Uploaded manifest to **QuickSight** and created visualizations.

Outcome:

Sentiment distribution visualized using bar and pie charts in **QuickSight**.

Final Output

Data:

- **Cleaned and translated reviews** stored in **S3**.
- **Sentiment-labeled results** uploaded to **S3**.
- **Visualizations** created using **QuickSight**.

Conclusion:

This project implemented an automated end-to-end pipeline for sentiment analysis using **AWS Glue**, **Step Functions**, **Lambda**, and **QuickSight**. The **HuggingFace Transformers** model

was used for sentiment analysis, providing greater control. **Telegram bot** integration enabled remote execution, making the system user-friendly.

Bonus Part – Automation with Lambda, Step Functions, and Telegram Bot

Goal:

Automate sentiment analysis and allow remote execution via a **Telegram bot**.

Lambda Functions:

1. **fetch_and_analyze_reviews**: Fetches reviews and sends them for sentiment analysis.
2. **send_email_notification**: Sends an email upon analysis completion.

Step Functions:

- Orchestrates Lambda functions to automate the analysis and notification process.

Telegram Bot:

- Triggered the **Step Function** via a **POST** request.

Outcome:

The **Telegram bot** allowed remote triggering of the sentiment analysis, and the entire process was automated.

Final Thoughts:

The system was fully automated using AWS services, integrating **Step Functions**, **Lambda**, **SNS**, and **QuickSight**. The **Telegram bot** added remote accessibility, and **HuggingFace Transformers** allowed for customized sentiment analysis, providing valuable insights.