Report
AWS

Part 1 – Data Collection & Preprocessing

Goal:
Collect and prepare customer reviews for NLP sentiment analysis.

Tools Used:
Python, BeautifulSoup, Requests, Pandas, deep_translator (GoogleTranslator), Amazon S3

Steps Taken:

- Created S3 bucket: `zanggar/technodom_reviews/`
- Scraped product titles and user reviews from Technodom.kz using BeautifulSoup and Requests
- Translated reviews from Russian to English using `deep_translator`
- Cleaned the data: removed empty values, fixed encoding issues, ensured consistent formatting
- Uploaded the cleaned dataset (`technodom_reviews_cleaned_final.csv`) to S3 for processing

Outcome:
Final dataset uploaded to S3 for use in further AWS-based processing.

Part 2 – Data Preparation with AWS Glue

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

Steps Taken:

1. **Crawler and Athena Querying:**
    - Ran AWS Glue crawler: `technodom_reviews_crawler`
    - Issue: Athena detected headers as data and used generic column names (col0, col1, etc.)

Solution: Created a new table in Athena with proper column names using SQL:

```
 CREATE TABLE my_data_catalog.technodom_reviews_fixed AS

SELECT col0 AS original, col1 AS translated

FROM my_data_catalog.technodom_technodom_reviews

WHERE col0 != 'original';
```

    -

2. **ETL Job in AWS Glue:**
    - Developed a Glue job using PySpark to clean and deduplicate the data:
        - Dropped nulls and duplicates
        - Cast columns to correct types
        - Saved cleaned output to: `s3://zanggar/technodom_reviews/cleaned_final_output`
3. **Validation in Glue Notebook:**
    - Used boto3 and pandas to load cleaned data
    - Previewed and validated structure using PySpark

Outcome:
Cleaned, validated dataset ready for machine learning and analysis.

Part 3 – Sentiment Analysis Using Transformers

Goal:
Perform sentiment analysis using HuggingFace Transformers instead of AWS Comprehend.

Steps Taken:

1. **Model Setup:**

Used HuggingFace pipeline for sentiment analysis:

```
 from transformers import pipeline

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

    -

2. **Processing:**

    - Loaded cleaned data from S3

Applied the model to the `translated` column:

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

    -

3. **Exporting Results:**

    - Saved the results to a local CSV: `sentiment_labels.csv`
    - Uploaded the file to S3: `s3://zanggar/technodom_reviews/results/sentiment_labels.csv`

Sample Output:

translated	label
"Great phone!"	positive
"Didn’t get the gift box"	negative

Outcome:
Final sentiment-labeled dataset prepared for visualization in QuickSight.

Part 4 – Visualization in Amazon QuickSight

Goal:
Create visual summaries of review sentiments using QuickSight.

Steps Taken:

1. **Data Source:**
    - Only one file used: `sentiment_labels.csv`
    - No merging required
2. **Manifest File:**

Created `manifest.json` to define file location and format:

```
{
 "fileLocations": [
   {
     "URIs": ["s3://zanggar/technodom_reviews/results/sentiment_labels.csv"]
   }
 ],
 "globalUploadSettings": {
   "format": "CSV",
   "delimiter": ",",
   "textqualifier": "\"",
   "containsHeader": true
 }
}
```

    -

3. **Connecting to QuickSight:**
    - Uploaded the manifest
    - QuickSight parsed `translated` and `label` columns correctly
    - Built visualizations using bar and pie charts

Visualizations Created:

- Proportion of positive vs negative reviews
- Review sentiment distribution

Outcome:
Interactive sentiment insights successfully visualized in QuickSight.

Final Output

- Cleaned and translated reviews stored in S3
- Sentiment-labeled results saved and uploaded
- Visualizations built using QuickSight
- Graphics available in the PDF: `visual_2025-04-22T12_31_26.pdf`