Report

Capstone Project – Social Media Analytics

Ngoc My Nguyen – Kununu Web Scraper and Text Analysis with LLM

1. Motivation

Before applying for jobs, I always check the review sections on platforms like Kununu or Glassdoor to learn more about companies I'm interested in. However, I often find it difficult to systematically gather all the information I need, and I don't always have the time to read every single review. Additionally, searching for specific details, such as whether a company has a works committee, can be quite challenging.
With the advancements in Large Language Models (LLM) and their capabilities, like web search and deep research offered by providers such as OpenAI, Google, and Anthropic, it has now become much easier to address these challenges.

2. Goal

I want to create a Streamlit application for personal use with the following components:

- **Web Scraping**: Collect Kununu employee reviews for a specific company of interest.

- **Text Analysis using LLM**: Analyze and summarize the reviews across 13 different categories:

  1. Arbeitsatmosphäre - Work Atmosphere
  2. Image - Image
  3. Work-Life-Balance - Work-Life Balance
  4. Karriere/Weiterbildung - Career/Professional Development
  5. Gehalt/Sozialleistungen - Salary/Benefits
  6. Umwelt-/Sozialbewusstsein - Environmental/Social Awareness
  7. Umgang mit älteren Kollegen - Interaction with Older Colleagues
  8. Vorgesetztenverhalten - Supervisor Behavior
  9. Arbeitsbedingungen - Working Conditions
  10. Kommunikation - Communication
  11. Gleichberechtigung - Equality
  12. Interessante Aufgaben - Interesting Tasks
  13. Sonstiges – Others

  Focus areas for analysis:

  - **Sentiment Analysis**: Since the reviews don't come with labels, I need the LLM to categorize them into positive and negative/critical aspects.
  - **Text Classification**: Extract additional insights from the three categories: *„Gut am Arbeitgeber finde ich"* (*„What I find good about the employer"*), *„Schlecht am Arbeitgeber finde ich" ("What I find bad about the employer")*, and *„Verbesserungsvorschläge" ("Suggestions for improvement")* and organize them into the 13 defined categories mentioned above.
  - **Similarity Analysis**: Group similar mentions across different reviews together for better organization.

- **Result Visualizations**:

  - **Wordcloud**: For a quick overview of topics.
  - **Treemap**: To visualize grouped mentions.

- **Raw Data Section**: A separate area to review and filter the scraped data for reference.

This project is primarily for personal use and tailored to my specific requirements. My main goal is to experiment with different options to assess whether they provide valuable insights. More importantly, it serves as an opportunity for me to learn and practice applying the concepts I've studied.

However, such applications could also be useful for organizations, such as HR departments. In that case, the methodology and implementation choices should be carefully selected and thoroughly assessed.

Note: In this project, both English and German are used because most Kununu reviews are in German. That's why I've asked the model to process the reviews in German. I could have used German for everything, but I want to keep the documentation in English as well. This mix of languages is intentional and, in my view, not a big issue.

3. How did I collect, process and analyze/model the data?

3.1. Data Collection

I collected the data through web scraping, focusing only on reviews from employees and ex-employees. The instructions are based on the first problem set I submitted before (I've included it again in the zip file for reference).

In that notebook, I originally wrote that I didn't run into any issues, but after taking a closer look, I found a problem. I later had added a section to scrape the labels (rating stars) for each category, but it didn't work. I tried several times with different approaches, but I still couldn't get the labels, so I decided to leave them out.

Each scraping result (as well as result from LLM analysis) is saved as a JSON file with a timestamp. This is important because I keep all the results in one folder, so each file needs a unique identifier.

3.2. Data Processing and Analysis

For analysis, I used only LLM, specifically the gemini-2.5-flash-preview-05-20 model with the free tier Gemini API. During development, I processed a lot of texts and often hit the API rate limit. Sometimes, the results were also cut off because of the maximum token limit. To work around this, I used four Google accounts with four different API keys, switching between them as needed.

At first, I used one large prompt to have the model process all categories at once, but I quickly realized this approach also reached the output token limit. So I changed the method: I created 13 separate prompts for the 13 categories, and each scraped file is processed up to 13 times, depending on the selection. Later, I combined all the results to create one big output. This approach works quite well.

Here is an example of the prompt. The bold parts will be adjusted for each category.

```
“# Review-Analyse für Unternehmensinsights

## Aufgabe
Analysiere die bereitgestellten Mitarbeiter-Reviews systematisch und extrahiere positive und kritische Punkte für die definierten Kategorien. Erstelle eine strukturierte JSON-Ausgabe (results.json) mit detaillierten Insights und Häufigkeitsanalysen.

## Kategorien für die Analyse
1. **Arbeitsatmosphäre** (Teamgeist, Kollegialität, Betriebsklima)

## Analyseprozess
1. **Initialisierung**: Beginne mit einer leeren results.json Struktur
2. **Iterative Analyse**:
   - Analysiere jeden Review einzeln
   - Prüfe für jeden extrahierten Punkt, ob er bereits in results.json existiert. Behandle dabei semantisch ähnliche Aussagen als denselben Punkt. Beispiel: "schlechte Bezahlung" = "niedriger Lohn" = "unterdurchschnittliches Gehalt"
   - **Falls vorhanden**: Erhöhe count um 1 und füge neue Referenz zu references hinzu
   - **Falls neu**: Erstelle neuen Punkt mit count = 1 und erster Referenz
3. **Vollständigkeit**: Fahre fort bis alle Reviews analysiert sind

## Diese Textabschnitte sollen ebenfalls analysiert werden
- "Gut am Arbeitgeber finde ich"
- "Schlecht am Arbeitgeber finde ich"
- "Verbesserungsvorschläge"

## Extraktionsregeln
1. **Spezifität**: Jeder Punkt sollte ein konkreter und eindeutiger Satz sein; die Aussagen der Reviews können in Standardtexte umformuliert werden.
2. **Präzision**: Beschreibe Punkte klar und verständlich
3. **Sprache**: Alle Sätze sind ausschließlich auf Deutsch zu verfassen; bei Bedarf übersetze sie.

## Wichtige Zusammenfassungsregeln
Behandle semantisch ähnliche Aussagen als denselben Punkt
###**Arbeitsatmosphäre:**
- **Positive Kollegen-Beschreibungen**: "Angenehme Kollegen", "Nette Kollegen", "Freundliche Kollegen", "Sehr freundliches Team" → "Freundliche und angenehme Kollegen"
- **Teamzusammenhalt-Varianten**: "Guter Teamzusammenhalt", "Sehr starker Zusammenhalt", "Hervorragende Teamzusammenarbeit" → "Starker Teamzusammenhalt und gute Zusammenarbeit"
- **Atmosphäre-Beschreibungen**: "Angenehme Arbeitsatmosphäre", "Gute Arbeitsatmosphäre", "Positive Atmosphäre" → "Angenehme und positive Arbeitsatmosphäre"
```

```
## Gewünschte JSON-Struktur
"arbeitsatmosphaere": {
 "positive_points": [
  {
   "point": "Sehr gute Teamarbeit und Kollegialität",
   "count": 3,
   "references": [
    {
     "review_id": Firmenname_1,
     "employee_type": "Ex-Angestellte/r oder Arbeiter/in",
     "field": "Finanz / Controlling",
    },
    {
     "review_id": Firmenname_13,
     "employee_type": "Angestellte/r oder Arbeiter/in",
     "field": "Vertrieb / Verkauf",
    },
    {
     "review_id": Firmenname_24,
     "employee_type": "Führungskraft",
     "field": "IT",
    }
   ]
  }
 ],
 "critical_points": [
  {
   "point": "Hoher Konkurrenzdruck zwischen Kollegen",
   "count": 2,
   "references": [
    {
     "review_id": Firmenname_7,
     "employee_type": "Angestellte/r oder Arbeiter/in",
     "field": "Forschung / Entwicklung",
    },
    {
     "review_id": Firmenname_18,
     "employee_type": "Ex-Führungskraft",
     "field": "Marketing / Produktmanagement",
    }
   ]
  }
 ]
},
```

Gib einfach den Inhalt der JSON-Datei aus. Füge keine Kommentare oder andere Sätze hinzu, da ich die Antwort direkt in die JSON-Datei schreiben werde.“

3.3. Data Visualization

For each category, a word cloud is created by first cleaning and filtering the input text. This means removing short words, numbers, and common German stopwords. The words that remain are then weighted by how often they appear, so the more frequent words show up larger. I use the WordCloud library to generate the cloud and apply a custom color scheme based on the category. The result is an image where the most important words stand out visually. The treemap shows the mentions, how often they appear, and also in which exact raw review each one is found. For this, I use Plotly.

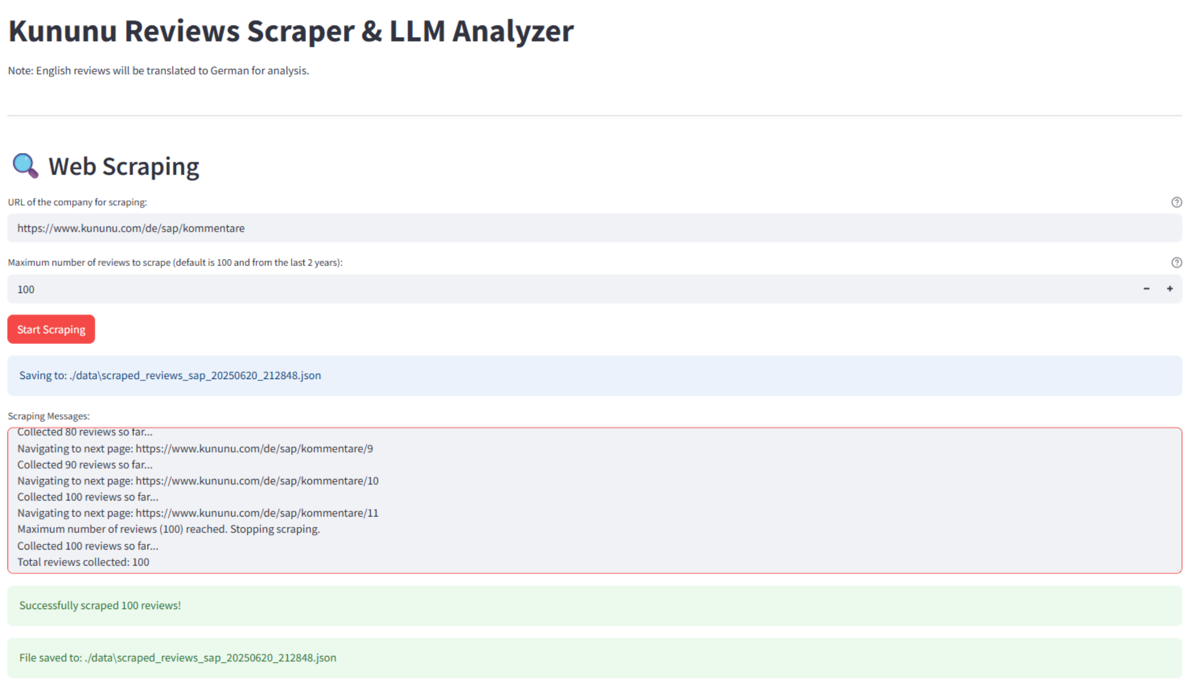
4. What are the results?

4.1. GUI

The Streamlit app, in my opinion, looks clean and visually appealing. It performed well during several test runs. The only drawback is that I didn’t implement caching.

4.2. Web Scraping

Apart from the issue with the star ratings, everything ran smoothly and efficiently. Below, is an example of the output response displayed in Streamlit.


Kununu Reviews Scraper & LLM Analyzer

Note: English reviews will be translated to German for analysis.

🔍 Web Scraping

URL of the company for scraping: https://www.kununu.com/de/sap/kommentare

Maximum number of reviews to scrape (default is 100 and from the last 2 years): 100

Start Scraping

Saving to: ./data\scraped_reviews_sap_20250620_212848.json

Scraping Messages:

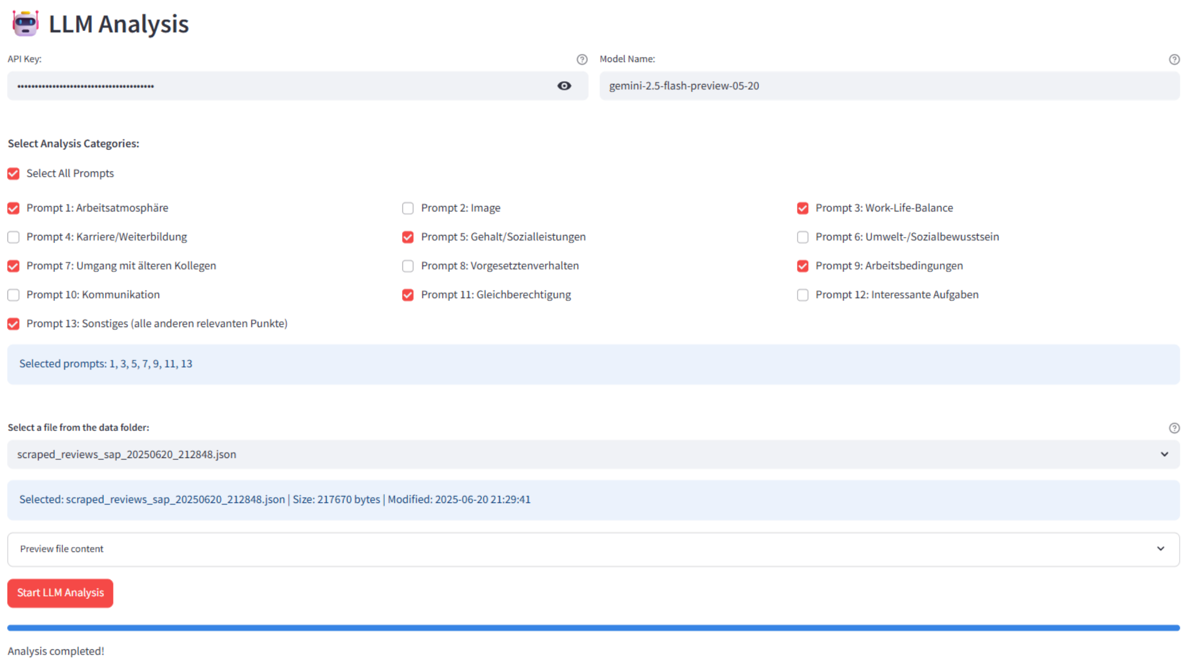
```
Collected 80 reviews so far...
Navigating to next page: https://www.kununu.com/de/sap/kommentare/9
Collected 90 reviews so far...
Navigating to next page: https://www.kununu.com/de/sap/kommentare/10
Collected 100 reviews so far...
Navigating to next page: https://www.kununu.com/de/sap/kommentare/11
Maximum number of reviews (100) reached. Stopping scraping.
Collected 100 reviews so far...
Total reviews collected: 100
```

Successfully scraped 100 reviews!

File saved to: ./data\scraped_reviews_sap_20250620_212848.json

4.3. LLM Analysis

The model handled the tasks defined in the prompts quite well. The figures below shows the output response as displayed in Streamlit.


LLM Analysis

API Key: ••••••••••••••••••••••••••••••••••••••

Model Name: gemini-2.5-flash-preview-05-20

Select Analysis Categories:

- [x] Select All Prompts

[x] Prompt 1: Arbeitsatmosphäre	[ ] Prompt 2: Image	[x] Prompt 3: Work-Life-Balance
[ ] Prompt 4: Karriere/Weiterbildung	[x] Prompt 5: Gehalt/Sozialleistungen	[ ] Prompt 6: Umwelt-/Sozialbewusstsein
[x] Prompt 7: Umgang mit älteren Kollegen	[ ] Prompt 8: Vorgesetztenverhalten	[x] Prompt 9: Arbeitsbedingungen
[ ] Prompt 10: Kommunikation	[x] Prompt 11: Gleichberechtigung	[ ] Prompt 12: Interessante Aufgaben
[x] Prompt 13: Sonstiges (alle anderen relevanten Punkte)		

Selected prompts: 1, 3, 5, 7, 9, 11, 13

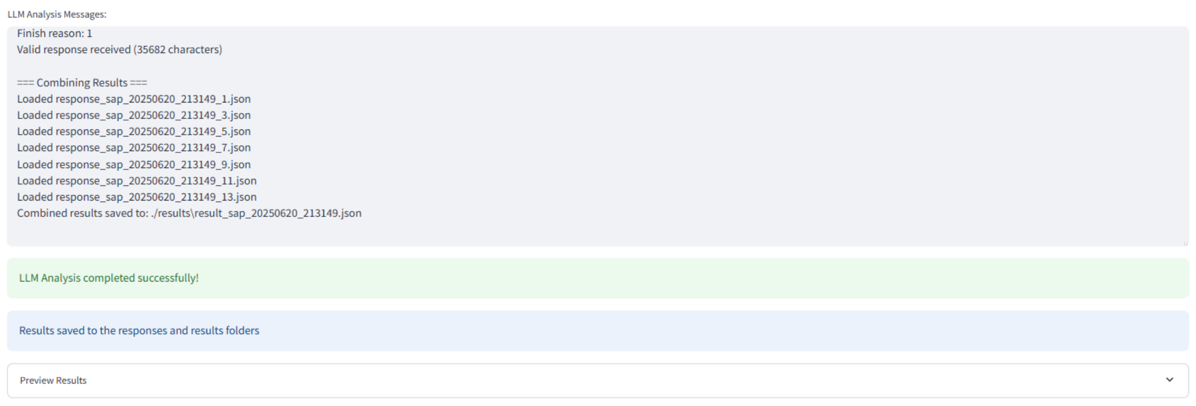
Select a file from the data folder: scraped_reviews_sap_20250620_212848.json

Selected: scraped_reviews_sap_20250620_212848.json | Size: 217670 bytes | Modified: 2025-06-20 21:29:41

Preview file content

Start LLM Analysis

Analysis completed!


LLM Analysis Messages:

```
Finish reason: 1
Valid response received (35682 characters)

=== Combining Results ===
Loaded response_sap_20250620_213149_1.json
Loaded response_sap_20250620_213149_3.json
Loaded response_sap_20250620_213149_5.json
Loaded response_sap_20250620_213149_7.json
Loaded response_sap_20250620_213149_9.json
Loaded response_sap_20250620_213149_11.json
Loaded response_sap_20250620_213149_13.json
Combined results saved to: ./results\result_sap_20250620_213149.json
```

LLM Analysis completed successfully!

Results saved to the responses and results folders

Regarding running time and interaction with the Gemini API: as the number of reviews the model analyzes in one prompt increases, the processing time grows proportionally. This also makes it more likely to run into rate limits, which can lead to more failures and retries.

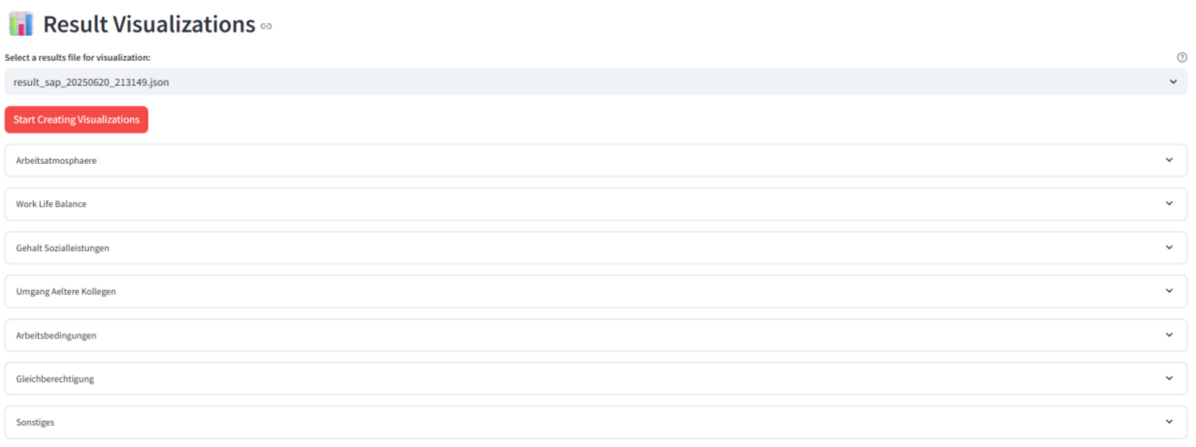
Below is a summary of the processing time and limitations. Detailed results for the scraped reviews or LLM responses can be found in the respective project folder.

Company	**Total scraped reviews**	**Scraping time**	**Categories analyzed**	**Analyzing time**	**Prompts with retry**	**Failed prompts (after 5 retries)**	**Tested with**
BMW Group	200	3 min	1-13	~ 2 hours	Prompt 2 (1 retry) Prompt 3 (3 retries) Prompt 4 (1 retry) Prompt 8 (1 retry) Prompt 9 (2 retries) Prompt 12 (4 retries) Prompt 13 (3 retries)	Prompt 1	Streamlit app
Bosch Gruppe	100	1.5 min	1-13	~ 1 hour	Prompt 4 (1 retry) Prompt 5 (1 retry) Prompt 9 (3 retries) Prompt 13 (1 retry)	no	Streamlit app
SAP SE	100	1.5 min	1, 3, 5, 7, 9, 11, 13	35 min	Prompt 1 (1 retry) Prompt 3 (1 retry) Prompt 13 (1 retry)	no	Streamlit app
Dräger	50	1.13 min	1, 3	6.5 min	no	no	Demo Jupyter Notebook

4.4. Results Visualizations

The visualizations are clear and help me quickly see the most important points for each topic. However, there are still quite a few duplicate points within the categories themselves that weren't grouped together, and some points in category 13 (Others) could actually fit into the defined 13 sections. This shows that the results from the model still need improvement, for example by refining the prompts.

The figures below shows the output response as displayed in Streamlit.



📊 Result Visualizations

Select a results file for visualization:

result_sap_20250620_213149.json

Start Creating Visualizations

Arbeitsatmosphaere

Work Life Balance

Gehalt Sozialleistungen

Umgang Aeltere Kollegen

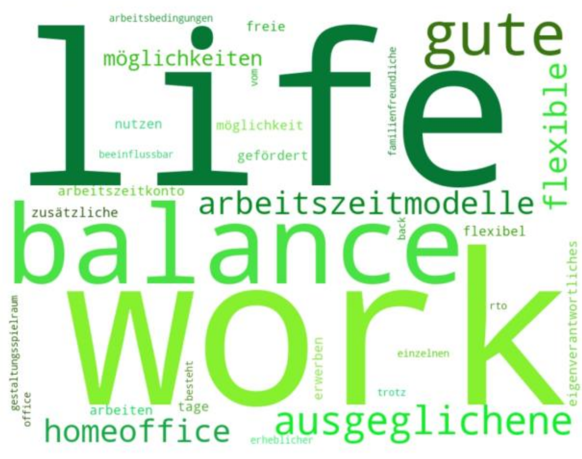
Arbeitsbedingungen

Gleichberechtigung

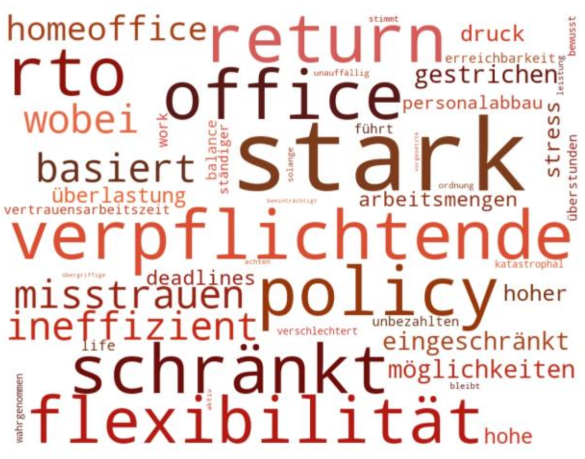
Sonstiges

Work Life Balance

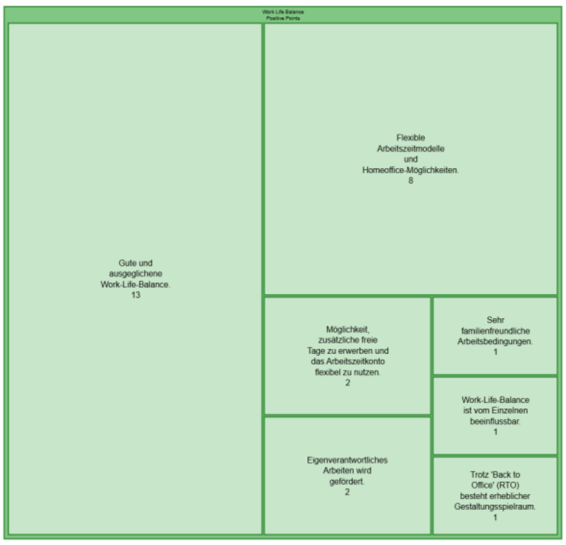
Wordcloud – Positive Points



Wordcloud – Critical Points



7 points • 28 total mentions


Work Life Balance
Positive Points

Gute und ausgeglichene Work-Life-Balance.
13

Flexible Arbeitszeitmodelle und Homeoffice-Möglichkeiten.
8

Möglichkeit, zusätzliche freie Tage zu erwerben und das Arbeitszeitkonto flexibel zu nutzen.
2

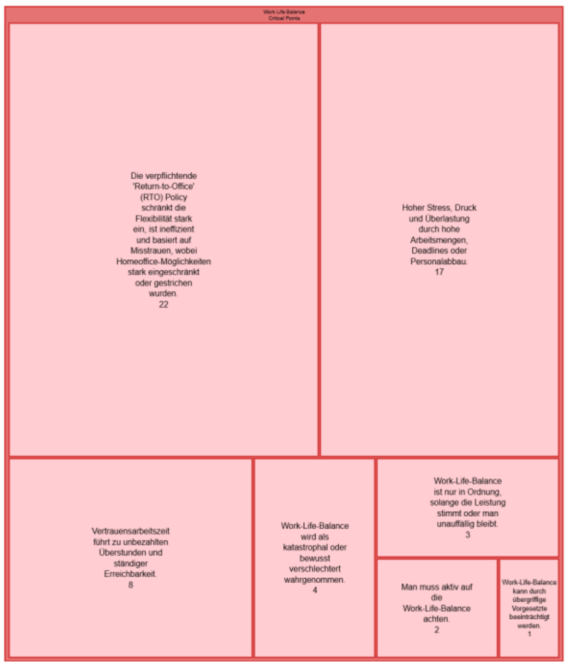
Sehr familienfreundliche Arbeitsbedingungen.
1

Work-Life-Balance ist vom Einzelnen beeinflussbar.
1

Eigenverantwortliches Arbeiten wird gefördert.
2

Trotz 'Back to Office' (RTO) besteht erheblicher Gestaltungsspielraum.
1

7 points • 57 total mentions


Work Life Balance
Critical Points

Die verpflichtende 'Return-to-Office' (RTO) Policy schränkt die Flexibilität stark ein, ist ineffizient und basiert auf Misstrauen, wobei Homeoffice-Möglichkeiten stark eingeschränkt oder gestrichen wurden.
22

Hoher Stress, Druck und Überlastung durch hohe Arbeitsmengen, Deadlines oder Personalabbau.
17

Vertrauensarbeitszeit führt zu unbezahlten Überstunden und ständiger Erreichbarkeit.
8

Work-Life-Balance wird als katastrophal oder bewusst verschlechtert wahrgenommen.
4

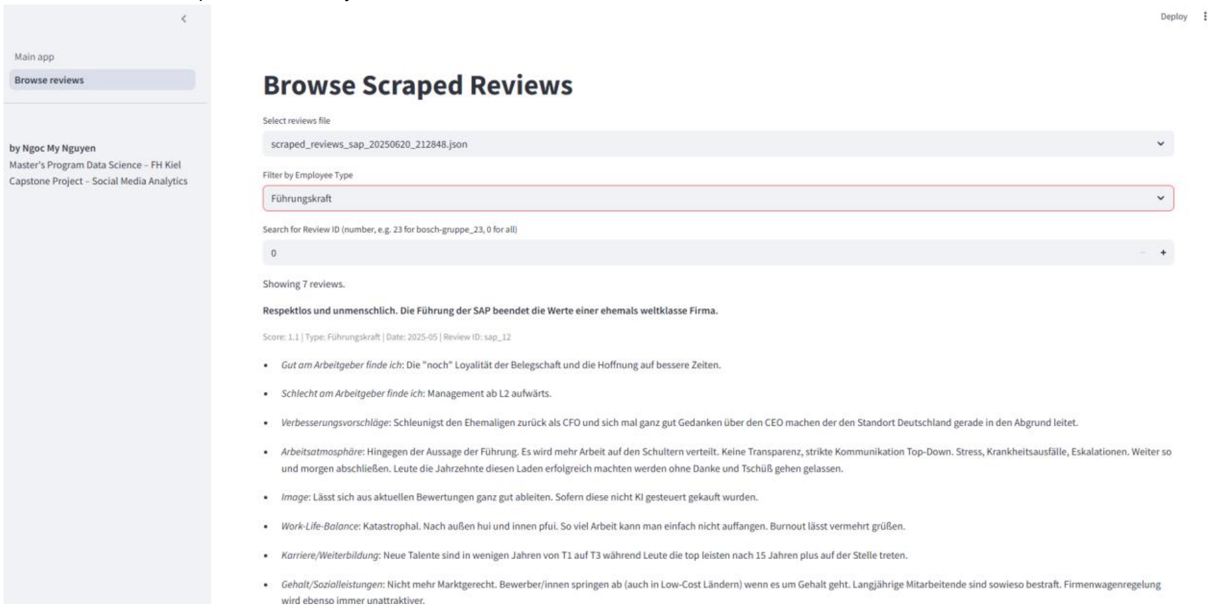
Work-Life-Balance ist nur in Ordnung, solange die Leistung stimmt oder man unauffällig bleibt.
3

Man muss aktiv auf die Work-Life-Balance achten.
2

Work-Life-Balance kann durch übergriffige Vorgesetzte beeinträchtigt werden.
1

4.5. Browse Scraped Reviews

I implemented this section as a separate page, which I can access through the sidebar. The options to filter by employee type and to search for a specific review by number also work well.



Main app

Browse reviews

by Ngoc My Nguyen
Master's Program Data Science – FH Kiel
Capstone Project – Social Media Analytics

Deploy

Browse Scraped Reviews

Select reviews file: scraped_reviews_sap_20250620_212848.json

Filter by Employee Type: Führungskraft

Search for Review ID (number, e.g. 23 for bosch-gruppe_23, 0 for all): 0

Showing 7 reviews.

Respektlos und unmenschlich. Die Führung der SAP beendet die Werte einer ehemals weltklasse Firma.

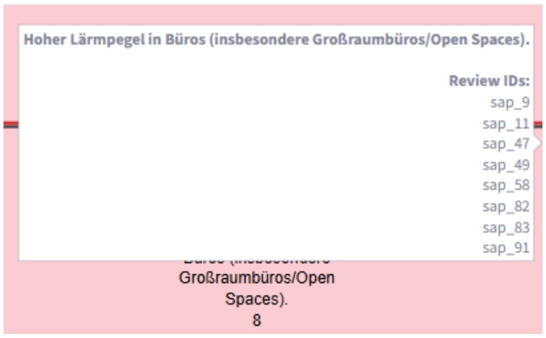
Score: 1.1 | Type: Führungskraft | Date: 2025-05 | Review ID: sap_12

- *Gut am Arbeitgeber finde ich*: Die "noch" Loyalität der Belegschaft und die Hoffnung auf bessere Zeiten.
- *Schlecht am Arbeitgeber finde ich*: Management ab L2 aufwärts.
- *Verbesserungsvorschläge*: Schleunigst den Ehemaligen zurück als CFO und sich mal ganz gut Gedanken über den CEO machen der den Standort Deutschland gerade in den Abgrund leitet.
- *Arbeitsatmosphäre*: Hingegen der Aussage der Führung. Es wird mehr Arbeit auf den Schultern verteilt. Keine Transparenz, strikte Kommunikation Top-Down. Stress, Krankheitsausfälle, Eskalationen. Weiter so und morgen abschließen. Leute die Jahrzehnte diesen Laden erfolgreich machten werden ohne Danke und Tschüß gehen gelassen.
- *Image*: Lässt sich aus aktuellen Bewertungen ganz gut ableiten. Sofern diese nicht KI gesteuert gekauft wurden.
- *Work-Life-Balance*: Katastrophal. Nach außen hui und innen pfui. So viel Arbeit kann man einfach nicht auffangen. Burnout lässt vermehrt grüßen.
- *Karriere/Weiterbildung*: Neue Talente sind in wenigen Jahren von T1 auf T3 während Leute die top leisten nach 15 Jahren plus auf der Stelle treten.
- *Gehalt/Sozialleistungen*: Nicht mehr Marktgerecht. Bewerber/innen springen ab (auch in Low-Cost Ländern) wenn es um Gehalt geht. Langjährige Mitarbeitende sind sowieso bestraft. Firmenwagenregelung wird ebenso immer unattraktiver.

5. What are the main insights and learning?

Overall, I was quite satisfied with the first results. Most importantly, I learned a lot about using API calls with LLMs and how to handle rate limits. I also learned how to create a Streamlit app, which was covered in Data Visualization course, but I hadn't had the chance to practice before.

The LLM performed quite well on the tasks I wanted. All functions worked as expected, but there is still room for improvement and further features, such as adding caching. Due to time constraints, I wasn't able to add references from the tree maps to the raw reviews.


Hoher Lärmpegel in Büros (insbesondere Großraumbüros/Open Spaces).

Review IDs:
sap_9
sap_11
sap_47
sap_49
sap_58
sap_82
sap_83
sap_91

You can try the app yourself by running **streamlit run Main_app.py**. You can either explore the provided data or scrape new reviews from other companies.

The required packages are: selenium, beautifulsoup4, streamlit, google-generativeai, and wordcloud.