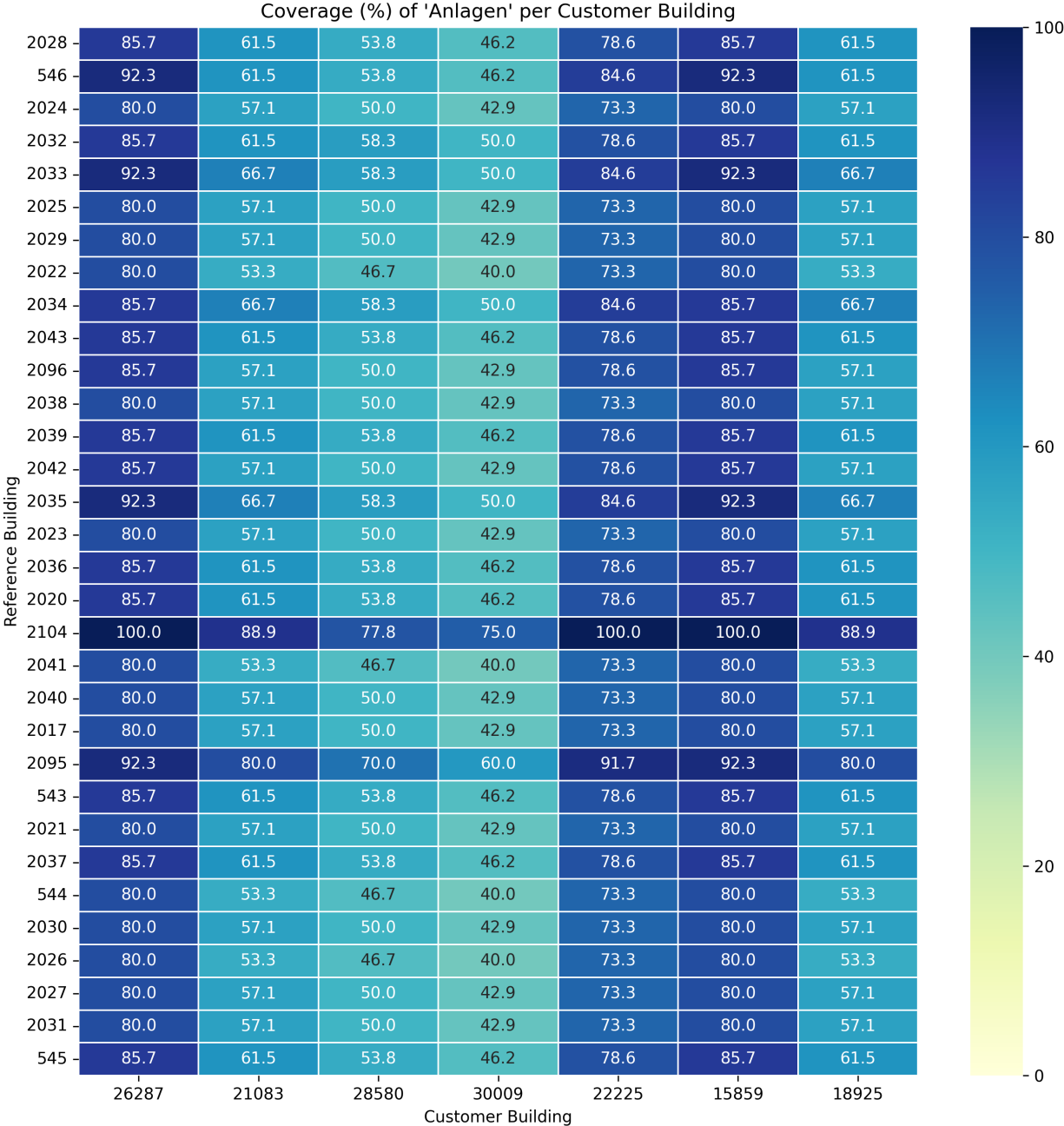


Ais Challenge

Aziz Becheur

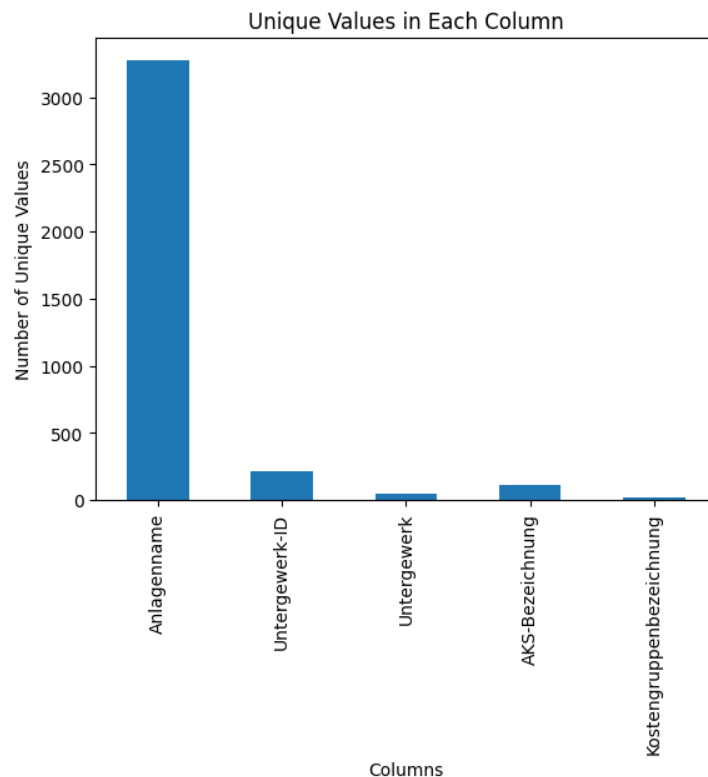
Vollständigkeitsprüfung:

Die Heatmap zeigt den Prozentsatz der Vollständigkeit jedes Kunden-Gebäudes im Vergleich zur Referenz



Data Exploration

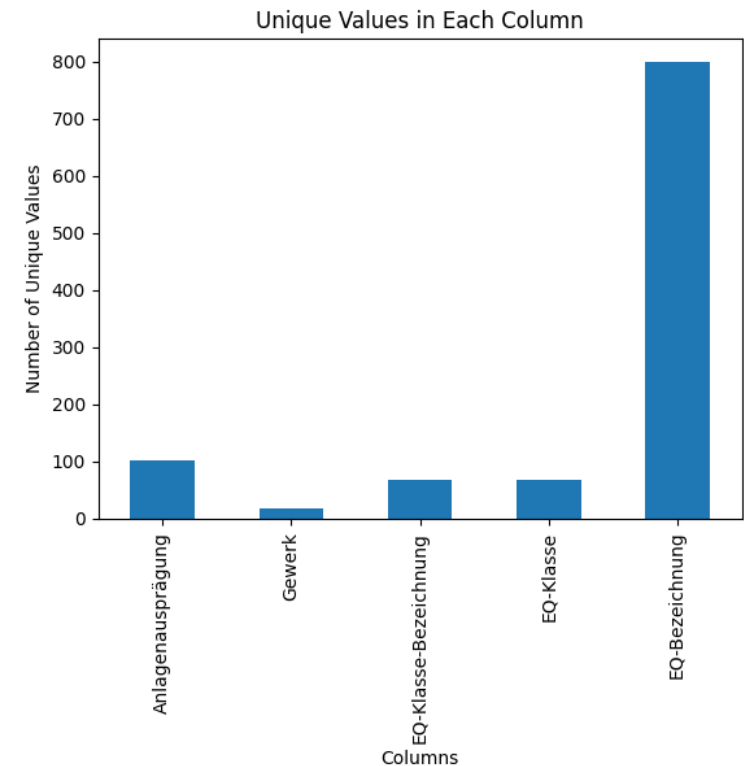
1. Die Kundendaten und Referenzdaten in einer hierarchischen Struktur organisieren, sodass:
Die Kostengruppenbezeichnung (entsprechende Kostengruppe als ID) einzigartige Unterkategorien von AKS-Bezeichnung (entsprechendes Untergewerk als ID) enthält und jede AKS-Bezeichnung wiederum in mehrere Untergewerk-IDs unterteilt ist.
2. In JSON Datei Speichern als Referenz
-> customer_data/jsons/ - reference_data/jsons
3. Jedes Gebäude sollte in einer eigenen CSV-Datei gespeichert werden, um sie miteinander vergleichen zu können.



Beispiel von Hierarchie:

Lufttechnische Anlagen

- └─ Kompressionskältemaschine
 - Kleinkaltwassersatz 109 kW
 - Kleinkaltwassersatz 237kW
 - VRV - Anlage
 - Kältemaschine Lüftungsanlagen
 - KM Kühlraum Küche
 - ...
- └─ Verteilnetz / Pumpen - Kälteerzeugung
 - Kältenetz Dach - Deckenkühl..
 - Kältenetz - Lüftungsgeräte



Lösungsweg:

Data Manipulation

4. Jede Mögliche Sequenz von Kategorie (KostenGruppenBezeichnung) + Unterkategorie (AKS-Bezeichnung) + Anlagen werden in einem separaten langen Text gespeichert und zu einem Vektor (Embedding) umwandeln.

(Gleich auch für Kundendatei)

-> 3_create_mapping.ipynb

5. **Semantischer Vergleich:** Die Nähe jedes **Embedding** zwischen Kundendatei und Referenz wird als ein Wert berechnet, den wir Cosine Similarity nennen. Diese Cosine Similarity zeigt, wie ähnlich sich zwei Einbettungen sind. (Check outputs/mapping.json)

-> 3_create_mapping.ipynb

6. Nested semantischer Vergleich, sodass wir Übereinstimmungen der Unterkategorien zweier übereinstimmender Kategorien finden.

-> 3_create_mapping.ipynb

7. Der verschachtelte semantische Vergleich hilft uns letztlich dabei, den EP-Katalog der Kundendatei zu erstellen. Jede übereinstimmende Kategorie/Unterkategorie mit den Referenzdaten erhält die entsprechende Referenz-ID zugewiesen.

RLT:

1. Lufttechnische Anlagen (score: 0.64)
2. Baukonstruktionen in Außenanlagen (score: 0.62)
3. Nutzungsspezifische Anlagen (score: 0.61)
4. Abwasser-, Wasser-, Gasanlagen (score: 0.60)
5. Neu Ausstattung (score: 0.59)

Elektro:

1. Baukonstruktionen in Außenanlagen (score: 0.63)
2. Neu Ausstattung (score: 0.58)
3. Gebäudeautomation (score: 0.57)
4. Starkstromanlagen (score: 0.55)
5. Fernmelde- und informationstechnische Anlagen (score: 0.53)

```
{ '31211':  
  1. { '432' (score: 0.72)  
  2. { '439' (score: 0.71)  
  3. { '431' (score: 0.70)  
  4. { '433' (score: 0.65)  
  5. { '434' (score: 0.64)
```

```
{ '31221':  
  1. { '439' (score: 0.62)  
  2. { '431' (score: 0.60)  
  3. { '432' (score: 0.60)  
  4. { '434' (score: 0.55)  
  5. { '433' (score: 0.50)
```

Um die einzelnen Schritte der Lösung nachvollziehen zu können, folgen Sie bitte der Reihenfolge der Jupyter-Notebooks:

1. 1_preparation_ref.ipynb
2. 2_preparation_data.ipynb
3. 3_create_mapping.ipynb
4. 4_comparison.ipynb
5. 5_zuordnen.ipynb

Eine ausführlichere Beschreibung der einzelnen Schritte finden Sie ebenfalls in den jeweiligen Notebooks.