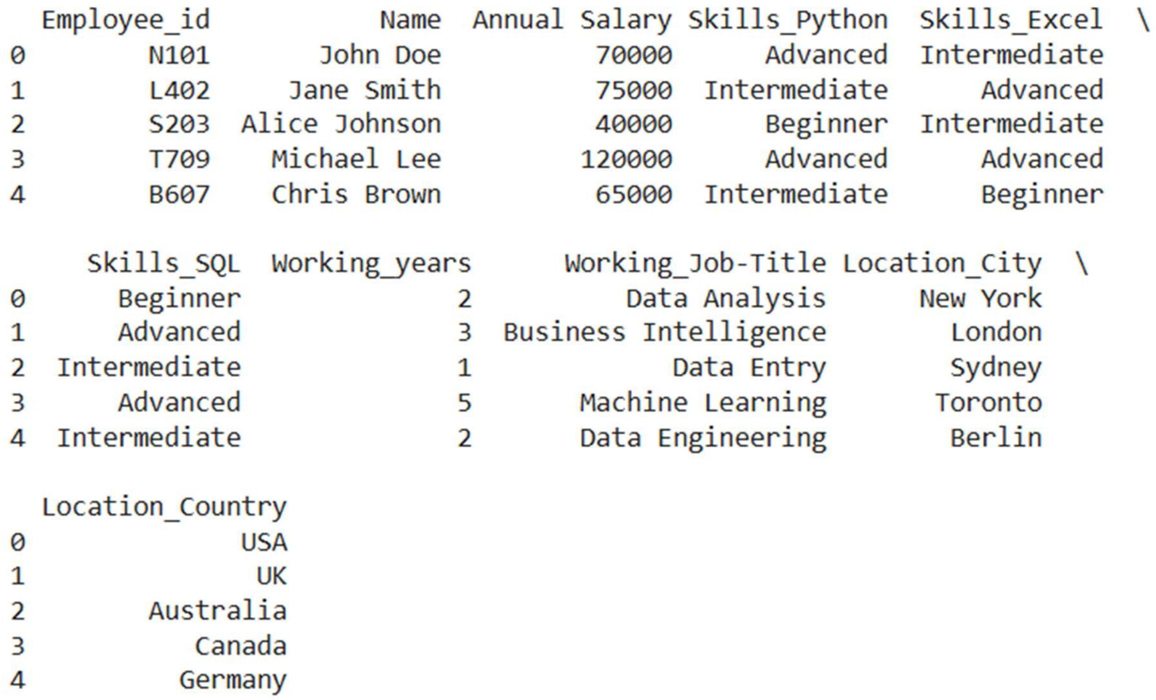
Create a pandas DataFrame from a JSON Object with Nested Structure

1. Task Description

The task involves extracting and transforming a nested JSON object into a structured format using Python's pandas library. The objective is to normalize the hierarchical JSON data into a flat pandas DataFrame while retaining meaningful relationships between fields through proper naming conventions. The JSON object contains information about employees, including details like their skills, job titles, locations, and annual salaries. By using pandas.json_normalize, the nested structure is flattened for easy analysis and further processing.

2. Screenshot of Output



```
  Employee_id          Name  Annual Salary Skills_Python  Skills_Excel  \
0        N101      John Doe          70000      Advanced  Intermediate
1        L402    Jane Smith          75000  Intermediate      Advanced
2        S203 Alice Johnson          40000      Beginner  Intermediate
3        T709   Michael Lee         120000      Advanced      Advanced
4        B607   Chris Brown          65000  Intermediate      Beginner

     Skills_SQL  Working_years      Working_Job-Title Location_City  \
0      Beginner              2          Data Analysis      New York
1      Advanced              3  Business Intelligence        London
2  Intermediate              1             Data Entry        Sydney
3      Advanced              5       Machine Learning       Toronto
4  Intermediate              2       Data Engineering        Berlin

  Location_Country
0              USA
1               UK
2        Australia
3           Canada
4          Germany
```

3. Algorithms Used in Task

a. Libraries Used:

1. pandas:

   - The pandas library is used for data manipulation and analysis.
   - Specifically, the json_normalize function was employed to flatten the nested JSON structure into a DataFrame.

2. json:

   - Used for representing the nested JSON data structure.

b. Methodology:

- Data Preparation: A JSON object representing employee details was created with nested structures for Skills, Working, and Location.
- Flattening JSON: The pandas.json_normalize function was applied to normalize the nested data. The 'sep' parameter was set to '_' to ensure hierarchical relationships are represented in column names.
- Output Validation: The resulting DataFrame is printed to validate the successful transformation of the nested JSON data.

c. Key Features:

- Nested fields like Skills, Working, and Location were flattened into columns with names such as 'Skills_Python' and 'Location_City'.
- Preserves the data integrity while transforming hierarchical data into a tabular format.

4. Report In Task Zip File

The task report has been added to the zip file. This includes:

1. The Python script (Data Science & Machine Learning_Task_2.ipynb).

2. A text version of this report (Task_2_Report.pdf).

Prepared by
Karan Gohil