Documentation: 4 Step - companies scoring

Overview

This script processes and evaluates company data from multiple sources to calculate a risk score for each company based on predefined criteria. The final risk level is assigned based on the calculated scores, and the results are saved to a CSV file.

Table of Contents

---

1. Introduction

The purpose of this script is to:

- Standardize and merge company data from multiple files.
- Filter companies from a specific list (Belgium-based companies).
- Aggregate scores from multiple evaluation steps.
- Classify companies into risk levels based on total scores.
- Export the final dataset with risk levels to a CSV file.

---

2. Dependencies

Ensure the following Python libraries are installed:

- `pandas` (for data manipulation and analysis)

Install dependencies if not already installed:

bash
Copy code
```
pip install pandas
```

---

3. Data Sources

The script processes four data files:

1. **BELGIUM_companies_short.xlsx**
    - Contains a list of Belgium-based companies with their names.
    - **Column Required:** `Name`
2. **Step_1_evaluated_companies.xlsx**
    - Company evaluation data from Step 1.
    - **Columns Required:** `Name`, `Score_Step_1`
3. **Step_2_company_status_report_with_scores.csv**
    - Company evaluation data from Step 2.
    - **Columns Required:** `OriginalCompanyName`, `Score`
4. **Step_3.2_company_analysis_with_scores.csv**
    - Company evaluation data from Step 3.
    - **Columns Required:** `company`, `score`

Expected File Formats:

- Excel files (`.xlsx`) for Step 1 and Belgium companies.
- CSV files (`.csv`) for Steps 2 and 3.

---

4. Data Processing Steps

Step 1: Load Data Files

- Load company data from four input files using `pandas.read_excel()` and `pandas.read_csv()`.

Step 2: Standardize Column Names

- Rename company name columns in each dataset to ensure consistency:
    - `Name`, `OriginalCompanyName`, and `company` → `company_name`

Step 3: Combine Score Data

- Merge scores from Step 1, Step 2, and Step 3 into a single dataframe (`all_scores`).

Step 4: Filter Belgium-Based Companies

- Filter the combined scores, keeping only companies present in `BELGIUM_companies_short.xlsx`.

Step 5: Aggregate Scores

- Group filtered data by `company_name` and calculate the total score for each company.

---

5. Risk Level Assignment Logic

A function `assign_risk_level(score)` assigns a risk level based on the total score:

- **Score > 30:** `prohibited`
- **7 <= Score <= 30:** `high`
- **1 <= Score <= 6:** `medium`
- **Score < 1:** `low`
- **Else:** `no risk`

This logic is applied to each company's total score.

---

6. Output

The final dataset includes:

- `company_name`: The standardized company name.
- `Score`: The total aggregated score.
- `risk_level`: The assigned risk level.

The dataset is saved as:

plaintext
Copy code
`Step_4_company_risk_scores.csv`

---

7. Usage Instructions

1. Place the required files (`.xlsx` and `.csv`) in the same directory as the script.

Run the script:
bash
Copy code
```
python script.py
```

2.
3. Upon completion, the output file `Step_4_company_risk_scores.csv` will be generated in the same directory.

---

8. Example Output (CSV Format)

company_name	Score	risk_level
Company A	35	prohibited
Company B	15	high
Company C	4	medium
Company D	0	low

---

9. Notes

- Ensure input file columns match the expected column names.
- Handle missing or incorrect data before running the script.

---

10. Future Improvements

- Add exception handling for file loading errors.
- Enable dynamic risk thresholds via configuration files.

Author: Natalja Krjuckova
Date: 24/12/2024

End of Documentation