Skills Required for Data Science Roles

By: Daniela Alejandra Gonzalez, Ikonkar Kaur Khalsa, Matthew Gregorio

Github Link

Summary

This package emphasizes the utilization of comprehensive sets of modular classes designed to pinpoint the essential skills necessary for data science roles. It leverages the Data Science Job Postings & Skills (2024) CSV available on Kaggle. The package identifies key skills for data science positions by analyzing the dataset and determines the most and least frequently mentioned skills in that very dataset. It also uncovers patterns and correlations suggesting proficiency in one skill may also require expertise in another skill for data science roles. By using the included classes and functions, users can efficiently import job skills data, count the most and least required individual and paired skills, and illustrate the most required.

Design

The package demonstrates a well-organized structure, featuring the following classes and their respective methods:

1. **CSVLoader:** This class streamlines the process of loading data from CSV files, providing a range of methods such as:
    a. **load_data()**: imports data from a CSV file
    b. **display_data():** showcases loaded data
    c. **tail_data():** previews the last few rows of data
    d. **head_data():** displays the first 5 rows of data
    e. **info():** comprehensive insights into the loaded dataset

    This method is great for data handling and exploration with CSV files since it makes it easy for users to load, view and understand the data.
2. **SkillCounter:** This class is responsible for analyzing data science skills that uses the following methods:
    a. **count_skills():** counts the most required individual skills and combined skills for data science roles
    b. **least_required_skills():** identifies the least required skills in the dataset

    This method counts the frequencies that then returns dictionaries with counts of individual skills, combined skills as tuples.
3. **WordCloudGenerator:** This class generates word clouds based on the top required skills. Its primary method is:
    a. **generate_wordcloud():** creates a visual representation of the 10 most required skills

    The text data are visually represented by the size of the words which will indicate its frequency and importance for data science roles.
4. **SkillsVisualizer:** This class focuses solely on creating visuals that represent combined skills combinations for data science roles by using this method:
    a. **visualize_chord_diagram():** generates an interactive chord diagrams representing skill interactions in an HTML format

    This method shows the relationships between data points. In this case, the diagram shows the relationship between different skill combinations. The width of the band between two skills represents the quantity of the co-occurrences. A wider band indicates more occurrences of pairing of skills.

Usage

There are several files that are incorporated into the final main.py file. The main file has imports of all important files including **CSVmodule**.py, **Counterskills.**py, **Visualizationmostrequired**.py, and **Visualizationmostrequiredcombination**.py.

1. **Data Loading:** An instance of the **CSVLoader** class is created, specifying the CSV file **("job_skills.csv")**. Data is loaded from the CSV file using the **load_data()** method.
2. **Skills Counting:** An instance of the **SkillCounter** class is created, which also specifies the CSV file **("job_skills.csv")**. The **count_skill()** method is called to count individual skills and skill combinations. The method then returns 3 results: **skill_counts**, **top_skills**, and **top_combinations**. Then the **least_required_skills()** method is called to retrieve the least required skills for data science roles.
3. **Printing Results:**

    a. **skill_counts:** counts of each skill
    b. **top_skills:** top 10 required skills
    c. **least_required:** 10 least required skills
    d. **top_combinations:** top 10 most common skill combinations
4. **Visualization:** Both **WordCloudGenerator** and **SkillsVisualizer** classes are used to create a word cloud for the top 10 required skills and top skill combinations in a chord diagram. The chord diagram is saved as an HTML file using **hv.save()**.

Discussion

This package complements general-purpose libraries like Pandas, Matplotlib and Seaborn by providing specialized functionality for analyzing job skills data and generating visualizations. While Pandas is a powerful library for data manipulation and analysis, this package focuses specifically on handling CSV files and a more targeted solution. Matplotlib and Seaborn are libraries that are commonly used for data visualization, however, this package includes specialized visualization modules tailored to showcase the most required skills and their combinations, which offers ready-to-use visualizations without needing any extensive customization. This package integrates seamlessly into data analysis pipelines focused on human resources, talent management, and workforce planning, which offers a quick and efficient solution for gaining insights into skill trends. This package could be improved by enhancing its flexibility by allowing users to customize visualization styles, incorporate additional data sources, or adjust counting methodologies. In this way it could increase its applicability to a wider range of usage. The package could also be improved by optimizing the performance that could then improve the package's scalability enabling it to handle larger datasets more efficiently.

Statement of Contributions

1. **Daniela Alejandra Gonzalez:** I developed the CSVLoader class, streamlining the process of loading data from CSV files and implementing various methods to enhance its functionality. Additionally, I played a key role in organizing the project and ensuring its availability on GitHub. Furthermore, I took ownership of completing the package repository, ensuring its readiness for distribution.
2. **Ikonkar Kaur Khalsa:** I contributed to the development of two essential classes: WordCloudGenerator and SkillsVisualizer, crucial for visualizing the data effectively. Collaborating closely with Matthew, I ensured the seamless printing of results through skill counting. Additionally, I took the lead in compiling the comprehensive project report and crafting an engaging presentation to showcase our findings.
3. **Matthew Gregorio:** I took the lead in conceptualizing the project proposal, laying the groundwork for its structure and defining its strategic direction. Additionally, I played a pivotal role in developing the SkillCounter class, implementing various methods crucial for accurate data calculation.

References

1. Continuum Analytics, Inc. et al. (2020). Bokeh: Python library for interactive visualization. https://bokeh.org/.
2. Hunter, J. D., Dale, D., Firing, E., Droettboom, M., & Lee, A. (2003). Matplotlib: A 2D Graphics Environment. https://matplotlib.org/.
3. McKinney, W., & others. (2020). pandas: Powerful data structures for data analysis. https://pandas.pydata.org/
4. Mueller, Andreas. (2019). Wordcloud: A little word cloud generator in Python (Version 1.8.1). https://github.com/amueller/word_cloud.
5. Rudiger, P., et al. (2020). HoloViews: Building complex visualizations easily. http://holoviews.org/.