Whether to keep integer columns in your dataset when you have a lot of float columns depends on several factors:

1. \*\*Data Precision and Range\*\*: If the integer columns represent categorical or discrete values where precision is not a concern (e.g., counts, IDs), keeping them as integers can be more efficient. However, if these integers could benefit from being in a floating-point format due to a range of values or precision needs, converting might be useful.

2. \*\*Memory Usage\*\*: Integers typically use less memory than floats. If memory usage is a concern and the integer columns don't require floating-point precision, keeping them as integers can be more efficient.

3. \*\*Data Analysis and Processing\*\*: Some analyses or operations might require a consistent data type across your dataset. If your algorithms or libraries work better with floats, or if you need to perform operations that mix data types frequently, converting integers to floats might simplify processing.

4. \*\*Data Storage and Compatibility\*\*: If you're storing the data in a database or using certain libraries that handle data types differently, you might need to adjust based on those requirements.

5. \*\*Future Proofing\*\*: Consider if the integer columns might need to accommodate a wider range or more precision in the future. If so, converting to floats might prevent future issues.

In summary, if integer columns are working well and are used in ways where their integer nature is beneficial, you might want to keep them as integers. If you face issues with data processing, memory, or need consistency with float columns, converting might be the better option.