To ensure the program detects metadata correctly after filtering the DataFrame, let's go through a few key components and ensure everything is properly structured:

1. **Metadata Detection**: Make sure the metadata is being detected from the filtered DataFrame after removing the ignored blocks.
2. **Handling Empty Values**: The code already masks empty values and re-applies them to the synthetic data.
3. **Writing Output**: Ensure the output maintains the original formatting, including the block IDs.

**Here’s a summary of how your code operates:**

1. **Load Layout**: Loads the layout from a CSV file and groups columns by Block\_ID.
2. **Load Sample Data**: Reads the sample data from a text file and stores the first and last rows for later use.
3. **Process Sample Data**: Processes each line of the sample data into a DataFrame according to the specified layout.
4. **Generate Synthetic Data**: Filters the DataFrame based on ignored blocks, detects metadata, fits the SDV model, and generates synthetic data.
5. **Evaluate Data**: Runs diagnostics and evaluates the quality of the synthetic data against the original.
6. **Write Output**: Writes the synthetic data to a new text file while maintaining the original structure.

**Additional Considerations**

* Make sure your layout.csv and generated\_file.txt files are properly formatted and contain the expected data.
* If you encounter errors related to missing metadata or column discrepancies, double-check that your ignored blocks are correctly specified and that the DataFrame retains the necessary columns for synthesis.
* You might want to implement error handling throughout the program to catch potential issues with file access or data format.

**Example Run**