

```

import pandas as pd

data = """brokered_by;status;price;bed;bath;acre_lot;street;city;state;zip_code;house_size;prev_sold_date
81824;sold;292677;5;3;0.29;63 Field Dr;Sumter;South Carolina;29154;3040;2021-12-22
... (rest of your data) ...
52946;for_sale;150000;4;2;10.0;675 Circle Dr;Hulbert;Oklahoma;74441;1896;2018-04-19"""

df = pd.read_csv(pd.compat.StringIO(data), sep=";")

# Deduplicate the DataFrame
deduped_df = df.drop_duplicates()

# Print the deduplicated DataFrame (optional - for verification)
print(deduped_df)

# Save the deduplicated DataFrame to a new CSV file (optional)
deduped_df.to_csv("deduped_dataset.csv", index=False, sep=";")

```

Explanation:

1. **Import pandas:** We use the pandas library for data manipulation.
2. **Read the data:** The provided data is read into a pandas DataFrame using `pd.read_csv`. The `pd.compat.StringIO` is used to treat the string data as a file-like object. The separator is specified as `sep=";"`.
3. **Deduplicate:** The `drop_duplicates()` method removes duplicate rows from the DataFrame. By default, it considers all columns when identifying duplicates. If you want to deduplicate based on a subset of columns, you can use the `subset` argument (e.g., `subset=['street', 'city', 'state', 'zip_code']`).
4. **Print and Save (Optional):** The code includes optional lines to print the deduplicated DataFrame to the console and save it to a new CSV file named `deduped_dataset.csv`. The `index=False` argument prevents the DataFrame index from being written to the file. The `sep=";"` ensures the output CSV uses the same separator as the input.

Key Improvements over simply using a set:

- **Handles mixed data types:** Pandas DataFrames can handle a mix of numeric, string, and date data types, which a simple set approach might struggle with.
- **Preserves data structure:** The output is a DataFrame, maintaining the original column names and data organization, making it easy to continue working with the data.
- **Flexibility:** Pandas provides more control over the deduplication process, allowing you to specify which columns to consider and how to handle missing values.
- **Efficiency:** For large datasets, pandas is generally more efficient than using Python's built-in set operations.

This improved code provides a robust and flexible way to deduplicate your dataset while preserving its structure and handling various data types.