1. Checking duplicates while ignoring some columns

If you want to check for duplicate records while ignoring specific columns, you can use the **duplicated** method along with the **subset** parameter. The **subset** parameter allows you to specify the columns that should be used to identify duplicates.

```

import pandas as pd

# Create a sample DataFrame

data = {'ID': [1, 2, 3, 1, 4],

'Name': ['Alice', 'Bob', 'Charlie', 'Alice', 'David'],

'Age': [25, 30, 35, 25, 40]}

df = pd.DataFrame(data)

# Check for duplicates while ignoring the 'Age' column

duplicate\_rows = df.duplicated(subset=['ID', 'Name'])

# Display the DataFrame with a new column indicating duplicate status

df['IsDuplicate'] = duplicate\_rows

# Display the DataFrame with duplicate status

print(df)

```

If you have a large number of columns to ignore, you can use the **difference** method to get the list of columns to check for duplicates.

```

columns\_to\_ignore = ['Age']

columns\_to\_check = df.columns.difference(columns\_to\_ignore)

duplicate\_rows = df.duplicated(subset=columns\_to\_check)

df['IsDuplicate'] = duplicate\_rows

print(df)

```