

In pandas, you can delete a particular column and remove duplicate rows using the following methods:

1. Delete a Particular Column

```
import pandas as pd

# Sample DataFrame
df = pd.DataFrame({
    'A': [1, 2, 2, 4],
    'B': [5, 6, 6, 8],
    'C': [9, 10, 10, 12]
})

# Drop column 'B'
df = df.drop(columns=['B'])

print(df)
```

2. Remove Duplicate Rows (Based on All Columns)

```
df = df.drop_duplicates()
```

3. Remove Duplicate Rows Based on Specific Columns

If you want to remove duplicates based on two specific columns (A and C in this case):

```
df = df.drop_duplicates(subset=['A', 'C'])
```

4. Keep First or Last Occurrence of Duplicates

By default, `drop_duplicates()` keeps the first occurrence. To keep the last one:

```
df = df.drop_duplicates(subset=['A', 'C'], keep='last')
```

To remove all duplicate rows (keeping none):

```
df = df.drop_duplicates(subset=['A', 'C'], keep=False)
```

Example

```
import pandas as pd
```

```
# Creating a DataFrame
```

```
data = {'A': [1, 2, 2, 3, 4, 4],  
        'B': [10, 20, 20, 30, 40, 40],  
        'C': [100, 200, 200, 300, 400, 400]}
```

```
df = pd.DataFrame(data)
```

```
# Deleting column 'B'
```

```
df = df.drop(columns=['B'])
```

```
# Removing duplicate rows based on columns 'A' and 'C'
```

```
df = df.drop_duplicates(subset=['A', 'C'], keep='first')
```

```
print(df)
```