mapper.py

```
#!/usr/bin/env python3
import sys


# Skip the header
for idx, line in enumerate(sys.stdin):
    if idx == 0:
        continue  # Skip header
    parts = line.strip().split(",")
    if len(parts) != 6:
        continue  # Skip malformed lines


    year = parts[0]
    try:
        max_temp = float(parts[3])
        min_temp = float(parts[4])
    except ValueError:
        continue  # Skip lines with non-numeric temperature


    # Emit key-value pairs
    print(f"{year}\t{max_temp},{min_temp},1")
```

reducer.py

```
#!/usr/bin/env python3
import sys
```

```
from collections import defaultdict

temp_data = defaultdict(lambda: [0, 0, 0])  # max_sum, min_sum, count

for line in sys.stdin:
    line = line.strip()
    if not line:
        continue
    parts = line.split("\t")
    if len(parts) != 2:
        continue
    year, values = parts
    try:
        max_temp, min_temp, count = map(float, values.split(","))
        temp_data[year][0] += max_temp
        temp_data[year][1] += min_temp
        temp_data[year][2] += count
    except ValueError:
        continue

# Output: Year -> avg max, avg min
for year in sorted(temp_data):
    max_sum, min_sum, count = temp_data[year]
    avg_max = max_sum / count
    avg_min = min_sum / count
    print(f"{year}\tAvg Max Temp: {avg_max:.2f}, Avg Min Temp: {avg_min:.2f}")
```

1. Open Terminal and switch to Hadoop user

```
pvg@pvg-HP-ProDesk-400-G4-SFF:~$ su hduser
Password:
```

2. Start HDFS

```
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ start-dfs.sh
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ start-yarn.sh
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ jps
```

3. Create an input directory

```
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ hdfs dfs -ls /
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ hdfs dfs -rm -r /input
#Similarly, delete any previous output files if present using: hdfs dfs -rm -r /output
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ hdfs dfs -mkdir -p /input
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ hdfs dfs -ls /
```

4. Create a text file, paste the weather data and upload it to HDFS

```
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ nano weather_data.txt
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ hdfs dfs -put weather_data.txt /input/
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ hdfs dfs -ls /input/
```

5. Similarly, create a mapper.py and reducer.py file

```
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ nano mapper.py
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ nano reducer.py
```

```
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ chmod +x mapper.py
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ chmod +x reducer.py
```

6. Run Hadoop streaming jar using the mapper and reducer scripts

```
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ whereis hadoop
hadoop: /usr/local/hadoop /usr/local/hadoop/bin/hadoop.cmd
/usr/local/hadoop/bin/Hadoop
```

```
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ hadoop jar
/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.3.4.jar \
> -input /input/weather_data.txt \
> -output /output/weather_output \
> -mapper mapper.py \
> -reducer reducer.py \
> -file mapper.py \
> -file reducer.py
```

7. View Output

```
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ hdfs dfs -ls /output/weather_output/
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ hdfs dfs -cat
/output/weather_output/part-00000
```

8. Stop HDFS

```
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ stop-dfs.sh
hduser@pvg-HP-ProDesk-400-G4-SFF:~$ stop-yarn.sh
```