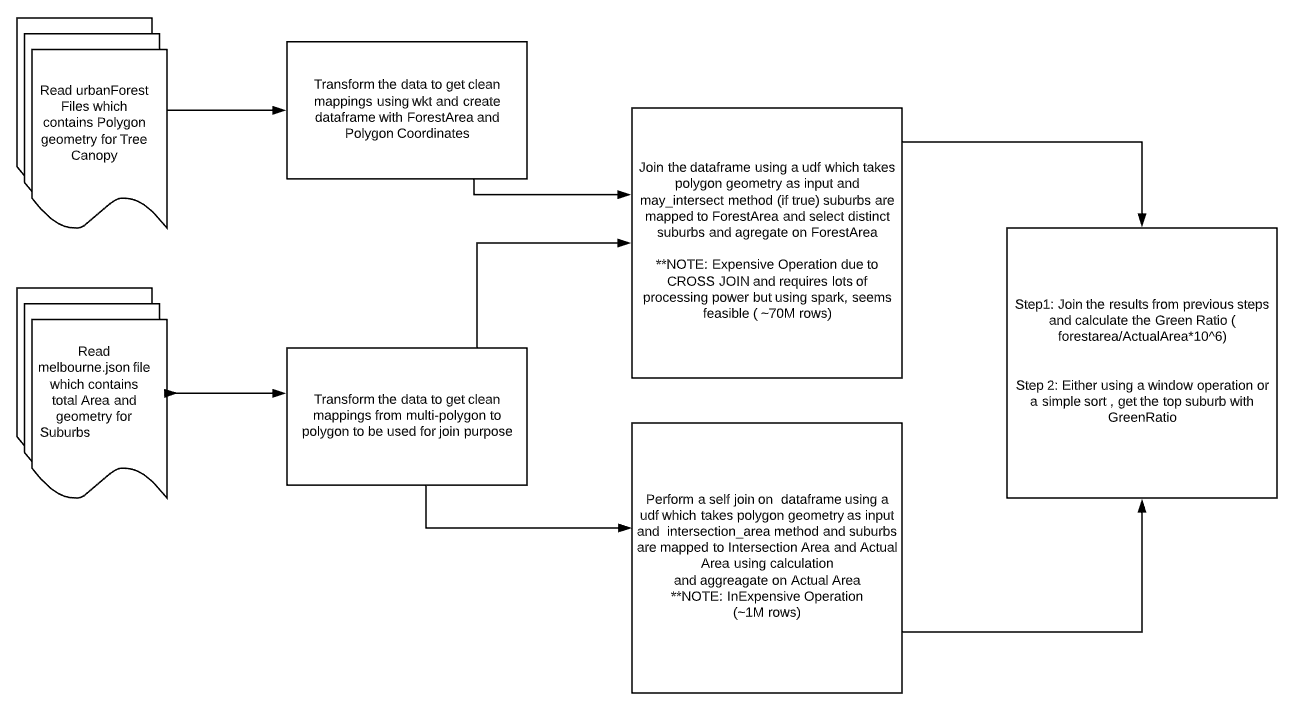
Approach Used:



This solution worked, however due to limited computing resources at my end( dual core) and 8GB, I find it very challenging to populate entire dataframe, hence it is used for a sample of tree canopy data against melb.json file.

***Alternate Approaches tested:***

1. Instead of doing a CROSS JOIN, collect the data but it results in memory overflow error. Hence discarded
2. Instead of CROSS JOIN polygon geometry from 2 dataframes, explode (long, lat) pairs from polygon list and perform an equality join.
   1. However, with this approach, accuracy is not good since, join happens on exterior.coordinates instead of holes and doesn’t provide correct forestArea after join leaving so many rows unjointed.
   2. If there is a method in shapely, which can match the holes coordinates to exterior coordinates, then it is the best solution and fastest but dropped due to inaccurate data
3. Saving the melbourne.json file into bucketed file by Suburbs (~42) and then performing cross join in parallel can yield in faster times. Dropped due to limited resource usage.

NOTE: Some of them are naïve approach, others are dependent on computing infrastructure and with only using pyspark (which is slow compared to Scala API), these are the best outcomes I have come across after testing it for a week or so.

Please let me know if you have any questions and I would love to know more about the correct approach if above doesn’t hold true to your expectation.