PART 2

1. Underrated Features of Go: -

a. Testing framework - Probably the most underrated feature of Go. To be honest, no one talks about it, but I'm in awe after using this simple yet effective framework. Write tests using the framework, put them in "_test" file, run "go test" and see the magic happen.
b. Simple yet powerful libraries - Developers often argue Go doesn't provide feature-rich libraries as in the case of Java and Python. I believe that is why Go is unique. If every feature is available in every language all of them will tend to be similar someday. Go was designed to be simple, that's why it only offers essential stuff with an efficient performance.
c. Speed - In my opinion, the speed of compilation is one of the underrated feature. What we don't realise is that this language performs better than other languages in the same segment. It was designed to be fast and this fact is often neglected.

2. Shoot in the foot: -

It’s still a young language that means from not supporting generic functions to no GUI library, Golang will take some time to match with it's older peers. For eg:- Go has no GUI library which means we need to invest a lot of time and knowledge to connect a library to our app instead of using native solutions like with Python or Java.

3. Unsafe and it’s use cases: -

As the name suggests, unsafe package is not safe to use. But Why? Because it gives us direct access to the memory and we can use it at our own will which is definitely against the design of Golang. Given that, there are still some use cases where unsafe usage can be justified:-

a. At the runtime when Go has to perform some memory operations such as allocating and freeing the stack memory it'll be useful to use unsafe as Go will be acting directly on the memory and unsafe will support to make such operations.
b. Unsafe usage in the sync package can also be justified. Let's take the example of sync.pool where pools are shared by goroutines via a part of memory, in such cases different goroutines or processors can access the memory using unsafe. Using a single pointer to access the full memory segment is definitely an effective way of designing pools.