The second method, Task.Run, has been introduced in a later version of the .NET framework (in .NET 4.5).

However, the first method, Task.Factory.StartNew, gives you the opportunity to define a lot of useful things about the thread you want to create, while Task.Run doesn't provide this.

For instance, lets say that you want to create a long running task thread. If a thread of the thread pool is going to be used for this task, then this could be considered an abuse of the thread pool.

One thing you could do in order to avoid this would be to run the task in a separate thread. A newly created thread *that would be dedicated to this task and would be destroyed once your task would have been completed.* You *cannot* achieve this with the Task.Run, while you can do so with the Task.Factory.StartNew, like below:

Task.Factory.StartNew(..., TaskCreationOptions.LongRunning);

As it is stated [here](https://devblogs.microsoft.com/pfxteam/task-run-vs-task-factory-startnew/):

So, in the .NET Framework 4.5 Developer Preview, we’ve introduced the new Task.Run method. **This in no way obsoletes** Task.Factory.StartNew, **but rather should simply be thought of as a quick way to use** Task.Factory.StartNew **without needing to specify a bunch of parameters. It’s a shortcut.** In fact, Task.Run is actually implemented in terms of the same logic used for Task.Factory.StartNew, just passing in some default parameters. When you pass an Action to Task.Run:

Task.Run(someAction);

that’s exactly equivalent to:

Task.Factory.StartNew(someAction,

CancellationToken.None, TaskCreationOptions.DenyChildAttach, TaskScheduler.Default);