```python
from utils import *
from constants import *
from agent import *
import numpy as np

class Task:
    def __init__(
            self,
            x: float = None,
            y: float = None,
            task_capacity: int = 1,
            task_radius: float = 100
            ):
        x = np.random.random()*1000 if x is None else x
        y = np.random.random()*1000 if y is None else y
        self.pos = np.array([x, y])
        self.task_capacity = task_capacity
        self.task_radius = task_radius

    def sufficient_agents_in_radius(self, agents: list, invoke_calloff: bool = False):
        """
        Checks whether there are enough agents within the task's radius for it to be complete.
        Also invokes calloff from agents within task radius if specified.
        """
        num_agents_in_radius = 0
        for agent in agents:
            # Checking if the agent is within the task radius, adding to num_agents_in_radius
            if distance_euclid(self.pos, agent.pos) < self.task_radius:
                num_agents_in_radius += 1
                agent.inside_task_radius = True

            # Checking if enough rgents are close enough to task to complete it
            if num_agents_in_radius >= self.task_capacity:
                for agent in agents:
                    if distance_euclid(self.pos, agent.pos) < self.task_radius:
                        agent.inside_task_radius = False
                        # Tells agents to perform calloff when this task is completed, if
specified
                        if invoke_calloff:
                            agent.calloff(agents)
                return True
        return False
```