

```
In [25]: from itertools import permutations

# from the article, change it as you wish
coalition_values = {
    frozenset(): 0,
    frozenset("1"): 1,
    frozenset("2"): 1,
    frozenset("3"): 1,
    frozenset(("1", "2")): 2,
    frozenset(("1", "3")): 2,
    frozenset(("2", "3")): 2,
    frozenset(("1", "2", "3")): 3,
}

def compute_shapley_values(coalition_values, player):
    players = max(coalition_values, key=lambda x: len(x))
    contributions = []

    for permutation in permutations(players):
        player_index = permutation.index(player)
        coalition_before = frozenset(permutation[:player_index]) # excluding player
        coalition_after = frozenset(permutation[:player_index + 1]) # player included
        contributions.append(coalition_values[coalition_after] - coalition_values[coalition_before])

    return sum(contributions) / len(contributions) # average, results in Shapley value

for player in ("1", "2", "3"):
    print("Player"+str(player)+" : " +str(compute_shapley_values(coalition_values, player)))

Player1 : 1.0
Player2 : 1.0
Player3 : 1.0
```

```
In [27]: from itertools import permutations

# from the article, change it as you wish
coalition_values = {
    frozenset(): 0,
    frozenset("1"): 1,
    frozenset("2"): 2,
    frozenset("3"): 3,
    frozenset(("1", "2")): 3,
    frozenset(("1", "3")): 4,
    frozenset(("2", "3")): 5,
    frozenset(("1", "2", "3")): 6,
}

def compute_shapley_values(coalition_values, player):
    players = max(coalition_values, key=lambda x: len(x))
    contributions = []

    for permutation in permutations(players):
        player_index = permutation.index(player)
        coalition_before = frozenset(permutation[:player_index]) # excluding player
        coalition_after = frozenset(permutation[:player_index + 1]) # player included
        contributions.append(coalition_values[coalition_after] - coalition_values[coalition_before])

    return sum(contributions) / len(contributions) # average, results in Shapley value

for player in ("1", "2", "3"):
    print("Player"+str(player)+" : " +str(compute_shapley_values(coalition_values, player)))
```

Player1 : 1.0

Player2 : 2.0

Player3 : 3.0

In []:

In []: