Exercise 1: The Shapley value

In this exercise, we implement the original Shapley value for a cooperative game. The game we consider here is an AI quiz, where we have five players: Timnit, Margret, Samy, Jeff, and Larry. The overall payoff function for a set of players S is given as

$$\text{payoff}(S) = 10t + 10m + 10s + 2j + 20(t \wedge m) + 20(t \wedge m \wedge s) - 30((t \vee m \vee s) \wedge j)$$

where t, m, s, j, l indicate whether Timnit, Margret, Samy, Jeff, and Larry are in the set S. The function `payoff(coalition)` is supposed to implement this functionality, where a set is a list/set of the respective characters (`'t', 's', 'm', 'j', 'l'`).

(a) Calculate the payoff of $S = \{$`t,m`$\}$ (result should be 40) and $S = \{$`t,j,s`$\}$ (result should be -8).

(b) Implement the original, exact Shapley value algorithm, using the set definition of Shapley values:

- Start by implementing the function `payoff(coalition)`, which should return the payoff of a subset `coalition` as given in the task. You can do this, for example, by first assigning a Boolean value for each player.
- Define a function `all_unique_subsets(population)` returning all subsets of a given set `population`.
- Finally, implement `shapley_set(member, population, v_function)` calculating the Shapley value for an arbitrary value function using the set definition.

Test your functions with the examples from (a) or with other examples from this game, i.p. corner cases like the empty set or the full set of all players.

(c) Analogously, implement the exact Shapley value algorithm using the order definition (or permutation definition) `shapley_perm(member, population, v_function)`. For this purpose, you can write (or use) a function that returns all orderings / all permutations of a given set. Again, test your function as in (b).

(d) Next, implement the permutation-based approximation discussed in the lecture: `shapley_perm_approx(member, population, v_function, num_iter)`. Here, `num_iter` is a fixed number of iterations given.

Note that because we implement the Shapley values for game theory, not yet for machine learning, we are not concerned with PD-functions and with data sets, but we can calculate the payoff function directly. This means that sampling is only required concerning the permutations, the sampling concerning the data points can be omitted.

Again, don't forget to test your function. Compare it to the function in (c) concerning runtime and accuracy for different numbers of iterations.

(e) In the last part of the exercise, we want to test whether the Shapley values or specific sets fulfill the axioms that uniquely define the Shapley values. Please write the functions `efficiency_check`, `symmetry_check`, `dummy_check`, and `additivity_check` to check the respective properties, all of which return a Boolean value. We briefly recall the properties below:

(i) *Efficiency:* Player contributions add up to the total payout of the game:

$$\sum\nolimits_{j=1}^{p} \phi_j = v(P).$$

(ii) *Symmetry:* Assess whether two features with the same contributions have the same Shapley value. I.e., for any two players $\{j, k\}$, if for any $S \subseteq P \setminus \{j, k\}$ we have:

$$v(S \cup \{j\}) - v(S) = v(S \cup \{k\}) - v(S),$$

then the Shapley values of j and k should be identical.

(iii) *Dummy:* Given a player j which makes no contribution for all subsets $S \subseteq P \setminus \{j\}$ (that is $v(S) = v(S \cup \{j\}))$, the Shapley value of this player must be zero.

(iv) *Additivity:* If v is the sum of two games v_1 and v_2 (i.e. $v(S) = v_1(S) + v_2(S)$ for any S), then the payout is the sum of payouts: $\phi_{j,v} = \phi_{j,v_1} + \phi_{j,v_2}$.

Again, test these functions and use them to check whether the axioms are fulfilled for our specific example.

Exercise 2: Axioms of Shapley Values

In this exercise, we want to mathematically prove that the Shapley values in cooperative game theory, as defined in the lecture, fulfill the four axioms discussed in the lecture. Throughout this exercise, we assume that the order and the set definition of Shapley values yield the same result, so you can always choose which definition to use.

(a) Prove that the symmetry, dummy player, and additivity axioms of Shapley values hold for any value function and any set P of players. These axioms are rather straightforward to prove, and this can be done using either of the definitions.

(b) **Bonus:** Prove the efficiency axiom for Shapley values.

Hint: This is easier when using the order definition. Plug the definition into the sum inside the axiom, then swap the sums and argue why almost all terms of the inner sum vanish.

(c) **Bonus:** Similarly to the additivity axiom, prove the following homogeneity axiom:

If $\alpha \in \mathbb{R}$ is a real number and v the payout function of a game, and we define $\tilde{v} := \alpha v$ as another payout function, then the Shapley value also scales accordingly, that is for all features $j \in P$ we have:

$$\phi_{j,\tilde{v}} = \phi_{j,\alpha v} = \alpha \phi_{j,v}.$$

Note: This axiom together with the additivity axiom shows that the Shapley value is linear in the payout function, so we have linearity of the Shapley value.