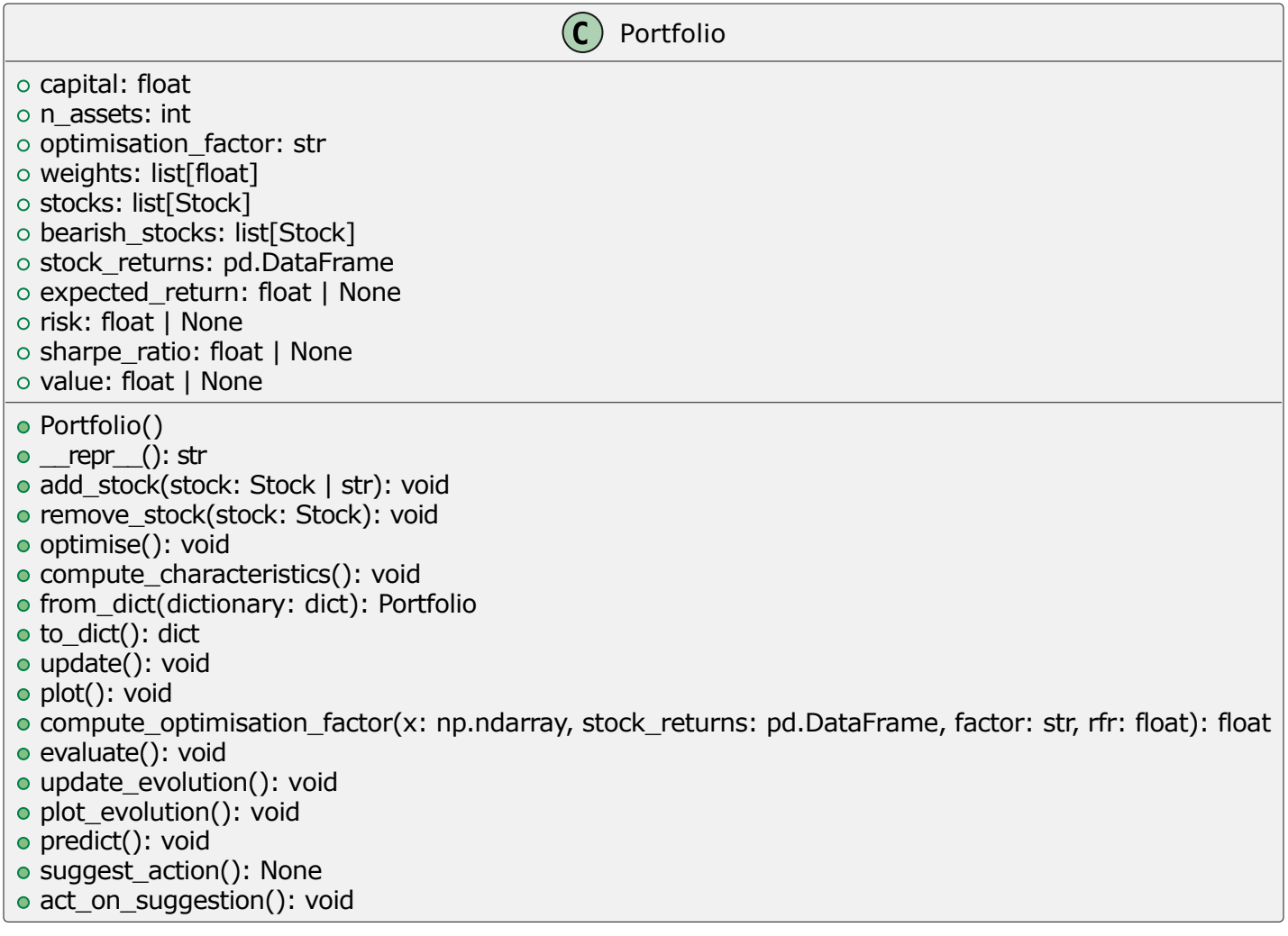
Portfolio
capital: float
n_assets: int
optimisation_factor: str
weights: list[float]
stocks: list[Stock]
bearish_stocks: list[Stock]
stock_returns: pd.DataFrame
expected_return: float | None
risk: float | None
sharpe_ratio: float | None
value: float | None
Portfolio()
__repr__(): str
add_stock(stock: Stock | str): void
remove_stock(stock: Stock): void
optimise(): void
compute_characteristics(): void
from_dict(dictionary: dict): Portfolio
to_dict(): dict
update(): void
plot(): void
compute_optimisation_factor(x: np.ndarray, stock_returns: pd.DataFrame, factor: str, rfr: float): float
evaluate(): void
update_evolution(): void
plot_evolution(): void
predict(): void
suggest_action(): None
act_on_suggestion(): void