**КР 6**

1. Методи агрегації даних (bagging) при налаштуванні класифікаторів.

2. Перевірити, чи належить ядерна функція k(x, y) = exp(x + y^2) - ln(tg(x/y) + 2) до класу ядерних функцій Мерсена. Перевірку виконати на множині X={2,4,8,6,4,1}, Y = {4,6,8,5,3,1}. Аргументи тригонометричних функцій задано в радіанах.

1. это метаалгоритм композиционного обучения машин, предназначенный для улучшения стабильности и точности алгоритмов машинного обучения, используемых в статистической классификации и регрессии. Алгоритм также уменьшает дисперсию и помогает избежать переобучения. Хотя он обычно применяется к методам обучения машин на основе деревьев решений, его можно использовать с любым видом метода. Бэггинг является частным видом усреднения модели.

Если задан стандартный тренировочный набор D размера n, бэггинг образует m новых тренировочных наборов Di, каждый размером n′, путём выборки из D равномерно и с возвратом. При сэмплинге с возвратом некоторые наблюдения могут быть повторены в каждой Di. Если n′=n, то для больших n ожидается, что множество Di имеет (1 - 1/e) (≈63,2%) долю уникальных экземпляров из D, остальные будут повторениями. Этот вид сэмплинга известен как бутстрэп-сэмплинг. Эти m моделей сглаживается с помощью вышеупомянутых m бутстрэп-выборок и комбинируются путём усреднения (для регрессии) или голосования (для классификации).

Бэггинг ведёт к «улучшению для нестабильных процедур», в которые входят, например, искусственные нейронные сети, деревья классификации и регрессий и выбор подмножеств в линейной регрессии. Интересное применение бэггинга, показывающее улучшение в обработке изображений, показано в статьях Саху, Аплея и др. С другой стороны, метод может слегка ухудшить эффективность стабильных методов, таких как метод K-ближайших соседей

2.

x = np.array([2, 4, 8, 6, 4, 1])

y = np.array([4,6,8,5,3,1])

M = np.zeros((6, 6))

for i in range(6):

for j in range(6):

M[i][j] = np.exp(x[i] + y[j]\*\*2) - np.log(np.tan(x[i]/y[j]) + 2)

array([[6.56599682e+07, 3.18559318e+16, 4.60718663e+28, 5.32048241e+11,

5.98731168e+04, nan],

[4.85165194e+08, 2.35385267e+17, 3.40427605e+29, 3.93133430e+12,

4.42411579e+05, 1.47263277e+02],

[ nan, 1.28516001e+19, 1.85867175e+31, nan,

2.41549524e+07, nan],

[3.58491284e+09, 1.73927494e+18, 2.51543867e+30, 2.90488497e+13,

nan, 1.09609725e+03],

[4.85165194e+08, 2.35385267e+17, 3.40427605e+29, 3.93133430e+12,

4.42411579e+05, 1.47263277e+02],

[2.41549519e+07, 1.17191424e+16, 1.69488924e+28, 1.95729609e+11,

2.20256130e+04, 6.12002399e+00]])

def is\_pos\_def(x):

return np.all(np.linalg.eigvals(x) > 0)

false

NOT POSITIVE DEFINED MATRIX