Лабораторна робота № 4. Побудова ансамблів моделей класифікації та регресії засобами бібліотеки Scikit-Learn Python

Недашківська Н.І.

Для отримання максимальної оцінки потрібно виконати ВСІ етапи Ходу виконання роботи та оформити звіт. Звітом може бути, наприклад, файл jupyter notebook з кодом програми і текстовими поясненнями отриманих цифр відповідно до Ходу виконання роботи.

ЗВІТ МАЄ МІСТИТИ:

- результати по всім пунктам Ходу виконання роботи, в тому числі оцінки якості побудованих моделей,
- опис методу побудови ансамблю, який використовувався,
- **у текстовому вигляді пояснення отриманих цифр (!)**.

Бажано опрацювати матеріал https://scikit-learn.org/stable/modules/ensemble.html. За цим посиланням є **ОПИСИ МЕТОДІВ ПОБУДОВИ АНСАМБЛІВ**, які не увійшли до лекцій, приклади використання методів.

Захист роботи:

- Усно: демонстрація коду програми, яка реалізує завдання згідно з варіантом і Ходом виконання роботи. Відповіді на питання щодо коду програми, отриманих результатів та методів, які використовувалися у роботі.
- Відповідь на теоретичне питання, написання коду в редакторі пайтон за темою "Методи і алгоритми побудови ансамблів".

1 Хід виконання роботи:

1. Взяти дані з роботи № 2 за варіантом. 2D-дані представити графічно.
2. Розбити дані на навчальний, перевірочний та тестовий набори. Перевірочний набір використати для налаштування гіперпараметрів. Тестовий набір використати для остаточної оцінки якості моделей.

3. **Побудувати моделі нейронних мереж**, використовуючи класи

   MLPClassifier або **MLPRegressor** для класифікації / регресії згідно з варіантом.

   Багатошаровий персептрон чутливий до масштабування вхідних даних. Дані можуть бути приведені до діапазону $[0, 1]$ або $[-1, +1]$, або приведені до нульового середнього та одиничної дисперсії. Той самий метод масштабування має бути застосовано до перевірочних і тестових даних.

   В MLPClassifier або MLPRegressor спробувати підібрати найкращі значення гіперпараметрів з числа наведених нижче (згідно з варіантом), використовуючи решітчатий пошук.

   - Побудувати різні архітектури нейронних мереж шляхом варіювання значень параметру `hidden_layer_sizes`. Порівняти результати класифікації/ регресії, отримані на основі різних архітектур.
   - Використати різні методи розрахунку ваг (параметр `solver`); порівняти результати, отримані методами 'lbfgs', 'sgd' та 'adam'.
   - Використати різні функції активації для скритого шару; дослідити їх вплив на результат.
   - Дослідити вплив різних значень параметру регуляризації `alpha` класу MLPClassifier на результат класифікації.
   - Дослідити вплив різних методів визначення `learning_rate` на результат класифікації (для `solver`='sgd').
   - Розглянути різні значення `max_iter`.

   Використати `early_stopping=True`.

   Використати `warm_start=True`.

   Вивести значення функції втрат на декількох перших і декількох останніх ітераціях - у кожному варіанті.

   Вивести значення середньої точності класифікації/регресії на навчальній і тестовій множинах, використовуючи score - у кожному варіанті.

4. Виконати прогнози на основі моделей нейронних мереж.

5. Оцінити якість моделей нейронних мереж в задачах класифікації на основі: правильності (accuracy), матриці неточностей (confusion matrix), точності (precision), повноти (recall), міри F1 (F1 score), ROC-кривої, показника AUC.

   В задачах регресії вибрати найкращу модель за коефіцієнтом детермінації R^2, помилками MAE, MAPE, RMSE.

   Зробити висновки про якість роботи моделей нейронних мереж на тестових даних.

6. **Побудувати ансамблі моделей**, використовуючи наступні методи (згідно з варіантом !):

   - AdaBoostClassifier. Розглянути різні значення `n_estimators`, `learning_rate` та `algorithm`.
   - AdaBoostRegressor. Розглянути різні значення `n_estimators`, `learning_rate` та `loss`.
   - GradientBoostingClassifier. Розглянути різні значення `learning_rate`, `n_estimators`, `subsample`, `max_depth` та `max_features`.
     Перевірити гіпотезу, що `max_leaf_nodes` = k дає результати порівняні з `max_depth` = k-1, але значно швидше тренується.
   - GradientBoostingRegressor. Розглянути різні значення `loss`, `learning_rate`, `n_estimators`, `subsample`, `max_depth`, `max_features`.
   - BaggingClassifier. Розглянути різні значення `max_samples`, `bootstrap`, `n_estimators`.
   - BaggingRegressor. Розглянути різні значення `max_samples`, `bootstrap`, `n_estimators`.
   - RandomForestClassifier. Розглянути різні значення параметрів `max_depth` та `max_features`, `bootstrap`, `n_estimators`.
   - RandomForestRegressor. Розглянути різні значення параметрів `max_depth`, `max_features`, `bootstrap`, `n_estimators`.
   - ExtraTreesClassifier. Розглянути різні значення параметрів `max_depth`, `max_features`, `min_samples_split`.
   - ExtraTreesRegressor.
   - VotingClassifier. Розглянути різні значення `voting` та `weights`.
   - VotingRegressor. Розглянути різні значення `weights`.
   - StackingClassifier. Розглянути різні значення `final_estimator`, `stack_method`.
   - StackingRegressor. Розглянути різні значення `final_estimator`.

7. **Побудувати ансамблі Bagging, RandomForest, ExtraTrees, AdaBoost, GradientBoosting** (згідно з варіантом):

   - В якості `base_estimator` / `estimators` використати одну/ декілька моделей із параметрами по умовчанню: дерев рішень, логістичної регресії, svm тощо. Порівняти декілька ансамблів, які утворені на основі одних `base_estimator` /`estimators` і відрізняються значеннями гіперпараметрів.
     Спробувати підібрати значення гіперпараметрів, використовуючи решітчатий пошук.

- В задачах класифікації побудувати графіки залежності значень міри якості від значення `n_estimators` для ансамблів та індивідуальних моделей на одній координатній вісі. В якості міри якості можна обрати `accuracy_score`, F1 score або `zero_one_loss`. Графіки для індивідуальних моделей, очевидно, будуть горизонтальними прямими.

  В задачах регресії побудувати графіки залежності значень R^2, MAPE або RMSE від значення `n_estimators`.

- Оцінити якість ансамблю на основі прикладів oob (для ансамблів на основі беггінгу).

8. **Побудувати ансамблі Voting, Stacking** (згідно з варіантом) на основі найкращої моделі нейронних мереж з попереднього етапу роботи, та **найкращої моделі / моделей з роботи №2**.

   Порівняти як на заданих даних працюють ансамблі з жорстким і м'яким голосуванням.

   Спробувати підібрати значення гіперпараметрів ансамблів, використовуючи решітчатий пошук.

   Порівняти значення метрик якості ансамблів та окремих моделей, які утворюють ці ансамблі.

9. **Для всіх варіантів** в задачах класифікації навести приклад границі рішень `decision boundaries` на основі окремої моделі та ансамблю.

   В задачах регресії на одній координатній площині графічно навести:

   - прогнози на основі окремої моделі `base_estimator` / `estimators`,

   - пронози на основі ансамблю,

   - приклади з перевірочної / тестової множини даних.

10. Розрахувати **значення зміщення та дисперсії** для окремої моделі та ансамблю.

11. Що можна сказати про **час навчання** ансамблю порівняно з окремими моделями, які утворюють ці ансамблі ?

12. **Зробити висновки**. Чи мало місце перенавчання або недонавчання досліджених ансамблів ? Які гіперпараметри і яким чином треба налаштовувати для зменшення ефектів перенавчання /недонавчання ансамблю? Чи краще на тестових даних виконується ансамбль порівняно з індивідуальними моделями?

2 Варіанти завдань

1. ExtraTreesClassifier. Розглянути різні значення параметрів `max_depth`,

   `max_features` та `min_samples_split`. Порівняти час навчання ансамблю з і без ранньої зупинки.

Побудувати моделі нейронних мереж:

- Побудувати різні архітектури мереж шляхом варіювання значень параметру `hidden_layer_sizes`. Порівняти результати класифікації, отримані на основі різних архітектур.
- Розглянути різні значення `max_iter`, дослідити їх вплив на результат.

Додатково побудувати AdaBoostClassifier. До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

2. AdaBoostRegressor. Розглянути різні значення параметрів цього алгоритму. До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

   В моделях нейронних мереж:

   - побудувати різні архітектури мереж шляхом варіювання значень параметру `hidden_layer_sizes`; порівняти результати, отримані на основі різних архітектур,
   - розглянути різні значення `max_iter`, дослідити їх вплив на результат.

   Додатково побудувати RandomForestRegressor.

3. StackingRegressor. Розглянути різні значення параметрів `final_estimator`. До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

   В моделях нейронних мереж для регресії:

   - використати різні функції активації для скритого шару та значення `learning_rate`, дослідити їх вплив на результат,
   - розглянути різні значення `max_iter`, дослідити їх вплив на результат.

   Додатково побудувати ExtraTreesRegressor.

4. StackingClassifier. Розглянути різні значення параметрів `final_estimator` та `stack_method`.

   До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

   В моделях нейронних мереж:

   - дослідити вплив різних методів визначення `learning_rate` на результат класифікації (для `solver`='sgd'),
   - використати `warm_start=True`.

   Додатково побудувати ансамблі GradientBoostingClassifier.

5. AdaBoostClassifier. Розглянути різні значення параметрів цього алгоритму. До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

   Порівняти час навчання ансамблю з і без ранньої зупинки.

   В моделях нейронних мереж:

- Побудувати різні архітектури шляхом варіювання значень параметру `hidden_layer_sizes`. Порівняти результати класифікації, отримані на основі різних архітектур.
- Розглянути різні значення `max_iter`, дослідити їх вплив на результат.

6. BaggingClassifier. Розглянути різні значення параметрів `max_samples` та `bootstrap`.

   Побудувати моделі нейронних мереж:

   - Використати різні функції активації для скритого шару та різні значення `learning_rate`, дослідити їх вплив на результат класифікації.
   - Використати `early_stopping=True`.

   Додатково VotingClassifier.

7. VotingRegressor. Розглянути різні значення `weights`. До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

   В моделях нейронних мереж для регресії:

   - використати різні методи розрахунку ваг (параметр `solver`); порівняти результати, отримані методами: 'lbfgs', 'sgd', 'adam',
   - використати `warm_start=True`.

   Додатково BaggingRegressor. Розглянути різні значення `max_samples` та `bootstrap`.

8. RandomForestClassifier. Розглянути різні значення `max_depth`, `max_features`, `min_samples_split`.

   Побудувати моделі нейронних мереж:

   - Розглянути різні архітектури мереж шляхом варіювання значень параметру `hidden_layer_sizes`. Порівняти результати класифікації, отримані на основі різних архітектур.
   - Використати `early_stopping=True`.

   Додатково VotingClassifier.

9. GradientBoostingClassifier. Розглянути різні значення параметрів `learning_rate`, `subsample` та `max_features`. Порівняти час навчання ансамблю з і без ранньої зупинки.

   До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

   В моделях нейронних мереж:

   - Дослідити вплив різних значень параметру регуляризації `alpha` класу MLPClassifier, він зазвичай вибирається з діапазону 10 ** -np.arange(1,7), на результат класифікації.
   - Використати `warm_start=True`.

Додатково RandomForestClassifier.

10. StackingClassifier. Розглянути різні значення `final_estimator` та `stack_method`. До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - Побудувати різні архітектури шляхом варіювання значень `hidden_layer_sizes`. Порівняти результати класифікації, отримані на основі різних архітектур.
    - Використати `warm_start=True`.

11. VotingClassifier. Розглянути різні значення `voting` та `weights`.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - Побудувати різні архітектури мереж шляхом варіювання значень `hidden_layer_sizes`. Порівняти результати класифікації, отримані на основі різних архітектур.
    - Розглянути різні значення `max_iter`, дослідити їх вплив на результат.

12. GradientBoostingRegressor. Розглянути різні значення `learning_rate`, `subsample`, `max_features`, `loss`. Порівняти час навчання ансамблю з і без ранньої зупинки.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - використати різні функції активації для скритого шару, дослідити їх вплив на результат,
    - використати `warm_start=True`.

13. StackingClassifier. Розглянути різні значення `final_estimator` та `stack_method`. До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - дослідити вплив різних значень параметру регуляризації `alpha` класу MLPClassifier, він зазвичай вибирається з діапазону 10 ** -np.arange(1,7),
    - використати `early_stopping=True`.

14. VotingClassifier. Розглянути різні значення `voting` та `weights`.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - побудувати різні архітектури мереж шляхом варіювання значень `hidden_layer_sizes`; порівняти результати класифікації, отримані на основі різних архітектур,

- використати `early_stopping=True`.

Додатково RandomForestClassifier.

15. AdaBoostClassifier. Розглянути різні значення `learning_rate` та `algorithm`. Дослідити ансамблі, які включають моделі нейронних мереж та моделі класифікації, побудовані в роботі №2.

    Порівняти час навчання ансамблю з і без ранньої зупинки.

    В моделях нейронних мереж:

    - використати різні методи розрахунку ваг (параметр `solver`), порівняти результати класифікації, отримані різними методами: 'lbfgs', 'sgd', 'adam'.
    - розглянути різні значення `max_iter`, дослідити їх вплив на результат.

16. AdaBoostRegressor. Розглянути різні значення `learning_rate` та `loss`.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж для регресії:

    - Дослідити вплив різних методів визначення `learning_rate` на результат (для solver='sgd').
    - Використати `warm_start=True`.

    Додатково розглянути як працює BaggingRegressor.

17. BaggingClassifier. Розглянути різні значення `max_samples` та `bootstrap`.

    Побудувати моделі нейронних мереж:

    - дослідити вплив різних значень параметру регуляризації `alpha` класу MLPClassifier, він зазвичай вибирається з діапазону 10.0 ** -np.arange(1, 7), на результат класифікації,
    - використати `early_stopping=True`.

18. GradientBoostingRegressor. Розглянути різні значення `learning_rate`, `subsample`, `max_features`, `loss`.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші. Порівняти час навчання ансамблю з і без ранньої зупинки.

    В моделях нейронних мереж:

    - використати різні функції активації для скритого шару, дослідити їх вплив на результат,
    - використати `warm_start=True`.

    Додатково розглянути як працює BaggingRegressor.

19. AdaBoostClassifier. Розглянути різні значення `max_samples` та `bootstrap`.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - дослідити вплив різних значень параметру регуляризації `alpha` класу MLPClassifier, він зазвичай вибирається з діапазону 10.0 ** -np.arange(1, 7),
    - розглянути різні значення `max_iter`, дослідити їх вплив на результат.

    Додатково побудувати ансамблі RandomForestClassifier.

20. ExtraTreesClassifier. Розглянути різні значення `max_depth`, `max_features` та `min_samples_split`.

    Побудувати моделі нейронних мереж:

    - Дослідити вплив різних методів визначення `learning_rate` на результат класифікації (для solver='sgd').
    - Використати `early_stopping=True`.

    Додатково GradientBoostingClassifier.

21. StackingClassifier. Розглянути різні значення `final_estimator` та `stack_method`.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші. Порівняти час навчання ансамблю з і без ранньої зупинки.

    В моделях нейронних мереж:

    - Використати різні методи розрахунку ваг (параметр `solver`), порівняти результати класифікації, отримані різними методами: 'lbfgs', 'sgd', 'adam'.
    - Використати `warm_start=True`.

    Додатково RandomForestClassifier.

22. GradientBoostingRegressor. Розглянути різні значення `learning_rate`, `subsample`, `max_features`, `loss`. До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж для регресії:

    - використати різні методи розрахунку ваг (параметр `solver`), порівняти результати, отримані різними методами: 'lbfgs', 'sgd', 'adam',
    - використати `early_stopping=True`.

    Додатково побудувати ансамблі BaggingRegressor.

23. BaggingClassifier. Розглянути різні значення `max_samples` та `bootstrap`.

    Побудувати моделі нейронних мереж:

- Дослідити вплив різних методів визначення `learning_rate` на результат класифікації (для `solver`='sgd').
- Використати `early_stopping=True`.

Додатково побудувати ансамблі ExtraTreesClassifier. Розглянути різні значення параметрів `max_depth`, `max_features` та `min_samples_split`.

24. AdaBoostClassifier. Розглянути різні значення параметрів `learning_rate` та `algorithm`.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - Дослідити вплив різних значень параметру регуляризації `alpha` класу MLPClassifier, він зазвичай вибирається з діапазону 10.0 ** -np.arange(1, 7), на результат класифікації.
    - Використати `warm_start=True`.

25. BaggingClassifier. Розглянути різні значення параметрів `learning_rate` та `algorithm`.

    Побудувати моделі нейронних мереж:

    - Дослідити вплив різних методів визначення `learning_rate` на результат регресії (для solver='sgd').
    - Розглянути різні значення `max_iter`, дослідити їх вплив на результат.

26. GradientBoostingClassifier. Розглянути різні значення параметрів `learning_rate`, `subsample` та `max_features`. Порівняти час навчання ансамблю з і без ранньої зупинки.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - Побудувати різні архітектури нейронних мереж шляхом варіювання значень параметру `hidden_layer_sizes`. Порівняти результати.
    - Використати `warm_start=True`.

27. VotingClassifier. Розглянути різні значення параметрів `voting` та `weights`.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - Побудувати різні архітектури мереж шляхом варіювання значень параметру `hidden_layer_sizes`. Порівняти результати класифікації, отримані на основі різних архітектур.
    - Розглянути різні значення `max_iter`, дослідити їх вплив на результат.

28. GradientBoostingRegressor. Розглянути різні значення параметрів `learning_rate`, `subsample`, `max_features`, `loss`.

    До ансамблю включити моделі нейронних мереж, моделі з роботи №2 та інші.

    В моделях нейронних мереж:

    - Дослідити вплив різних значень параметру регуляризації `alpha` класу MLPClassifier, він зазвичай вибирається з діапазону 10.0 ** -np.arange(1, 7), на результат.
    - Використати `early_stopping=True`.

    Розглянути як працює ExtraTreesRegressor. Розглянути різні значення параметрів `max_depth`, `max_features` та `min_samples_split`.

29. RandomForestRegressor. Розглянути різні значення параметрів `max_depth` та `max_features`.

    Побудувати моделі нейронних мереж:

    - Дослідити вплив різних методів визначення `learning_rate` на результат регресії (для solver='sgd').
    - Розглянути різні значення `max_iter`, дослідити їх вплив на результат.

30. ExtraTreesClassifier. Розглянути різні значення параметрів `max_depth`, `max_features` та `min_samples_split`.

    Побудувати моделі нейронних мереж:

    - Дослідити вплив різних методів визначення `learning_rate` на результат (для solver='sgd').
    - Використати `early_stopping=True`.

3 Контрольні питання для захисту роботи

1. Поняття ансамбля моделей, слабкого і сильного учня.
2. Види ансамблів.
3. Що таке класифікатор з жорстким і м'яким голосуванням? Навести приклади.
4. Чому індивідуальні прогнозатори в ансамблі намагаються робити якомога більш несхожими між собою?
5. Що таке беггінг та бутстреп?
6. Якими є функції агрегування у бегінгу для задач класифікації та регресії?
7. Який зміст основних параметрів класу BaggingClassifier з sklearn.ensemble?

8. Який зміст основних параметрів класу BaggingRegressor з sklearn.ensemble?
9. Беггінг і вставка. Що у них спільного і в чому відмінність?
10. Що таке out-of-bag приклади і для чого вони використовуються?
11. Призначення `oob_score` і `oob_decision_function_` в BaggingClassifier.
12. Методи випадкових ділянок (random patches method) і випадкових підпросторів (random subspaces method).
13. Що таке випадковий ліс? За рахунок чого вноситься додаткова випадковість у дерева випадкового лісу?
14. Перетворення ознак в багатовимірний простір, використовуючи клас RandomTreesEmbedding.
15. Як оцінити значущість ознак за допомогою випадкового лісу?
16. Особливо випадкові дерева ExtraTrees. У чому їх переваги?
17. Переваги випадкового лісу.
18. Недоліки випадкового лісу.
19. Поняття та особливості бустингу.
20. Сутність та етапи алгоритму AdaBoost.
21. Як розраховується вага індивідуального прогнозатора в алгоритмі AdaBoost? Пояснити всі змінні у формулі для розрахунку цієї ваги.
22. Як розраховується зважена частота помилок індивідуального прогнозатора в алгоритмі AdaBoost? Пояснити всі змінні у формулі для розрахунку цієї частоти.
23. Як виконується оновлення ваг прикладів в алгоритмі AdaBoost? Пояснити всі змінні у формулі для оновлення ваг.
24. Як виконується прогнозування в алгоритмі AdaBoost? Пояснити всі змінні у формулі для розрахунку прогнозу.
25. Який зміст основних параметрів класу AdaBoostClassifier з sklearn.ensemble?
26. Який зміст основних параметрів класу AdaBoostRegressor з sklearn.ensemble?
27. Сутність та етапи алгоритму градієнтного бустингу.
28. Який зміст параметрів `max_depth`, `n_estimators`, `learning_rate` класу GradientBoostingClassifier з sklearn.ensemble?

29. Який зміст основних параметрів класу GradientBoostingRegressor з sklearn.ensemble?
30. Який зміст параметрів `warm_start`, `subsample` класу GradientBoostingClassifier з sklearn.ensemble?
31. Етапи виконання стекінгу.
32. Класи BaggingClassifier, BaggingRegressor.
33. Класи RandomForestClassifier, RandomForestRegressor.
34. Класи AdaBoostClassifier, AdaBoostRegressor.
35. Класи GradientBoostingClassifier, GradientBoostingRegressor.
36. Класи StackingClassifier, StackingRegressor.
37. Класи VotingClassifier, VotingRegressor.
38. ExtraTreesClassifier, ExtraTreesRegressor.