МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
Національний технічний університет   
«Харківський політехнічний інститут»  
Кафедра ГМКГ

**Лабораторна работа №2**

З дисципліни «Інтелектуальний аналіз даних»

Виконав:

Студент групи ІКМ-220 г.

Ульянов Кирило Юрійович

Перевірив:

Доц. Дашкевич А.О.

Харків 2023

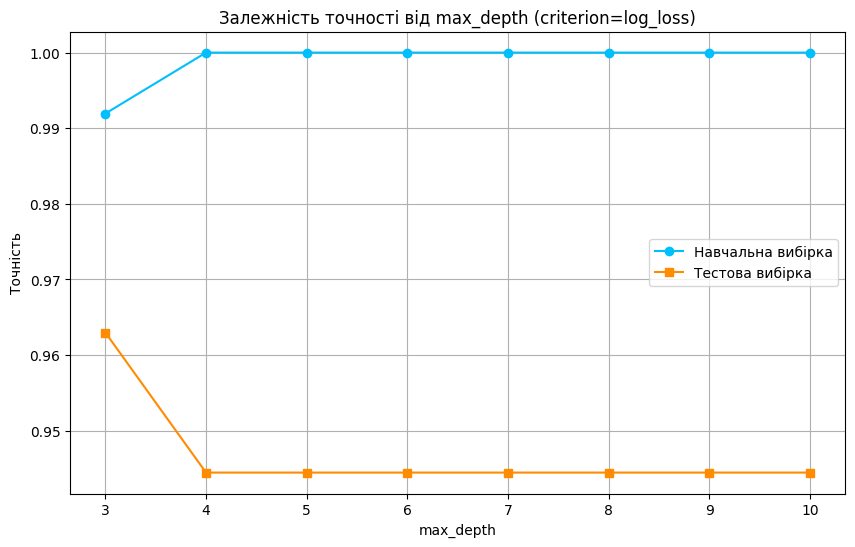
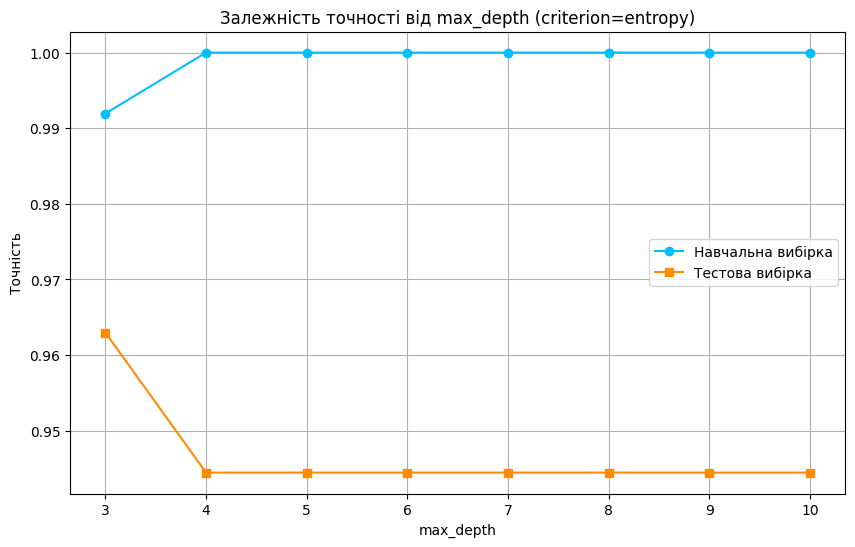
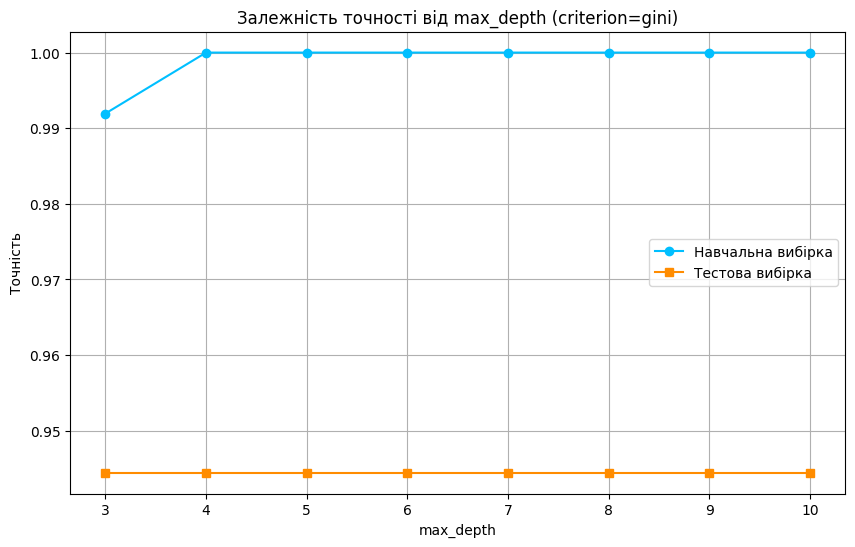
**Мета роботи:** вивчення застосувань дерев прийняття рiшень та їх ансамблю для класифiкацiї.

**Завдання на роботу:** завантаження набору даних, формування навчальної та тестової вибiрок, визначення функцiоналу якостi класифiкатору, вiзуалiзацiя результатiв.

Значення параметру train\_size = 0.7

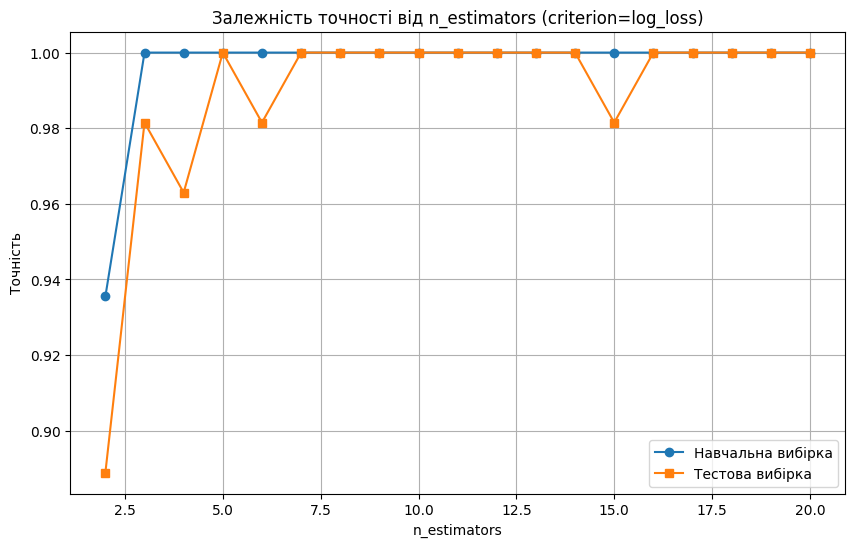
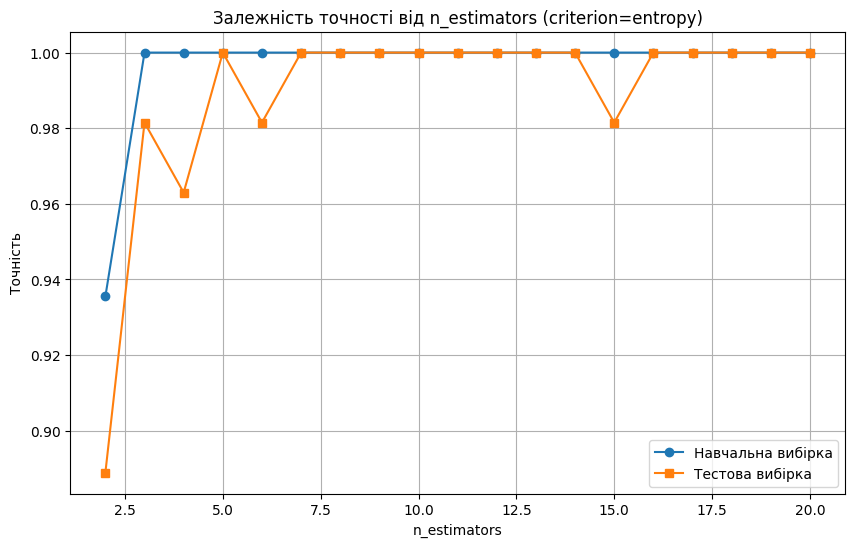
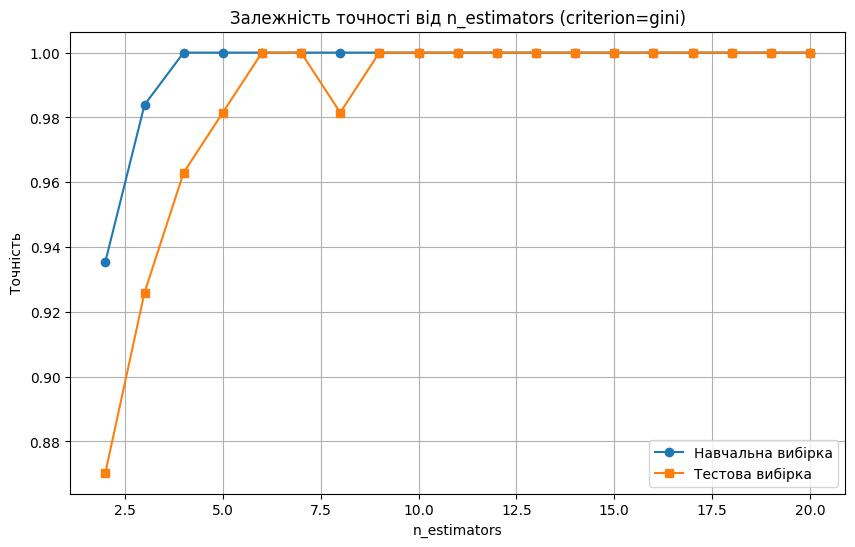
*Таблиця результатів дерева прийняття рішень*

|  |  |  |  |
| --- | --- | --- | --- |
| Критерій розбиття | max\_depth | Точність на навчальній вибірці | Точність на тестовій вибірці |
| gini | 3 | 0.992 | 0.944 |
| gini | 4 | 1.000 | 0.944 |
| gini | 5 | 1.000 | 0.944 |
| gini | 6 | 1.000 | 0.944 |
| gini | 7 | 1.000 | 0.944 |
| gini | 8 | 1.000 | 0.944 |
| gini | 9 | 1.000 | 0.944 |
| gini | 10 | 1.000 | 0.944 |
|  | | | |
| entropy | 3 | 0.992 | 0.963 |
| entropy | 4 | 1.000 | 0.944 |
| entropy | 5 | 1.000 | 0.944 |
| entropy | 6 | 1.000 | 0.944 |
| entropy | 7 | 1.000 | 0.944 |
| entropy | 8 | 1.000 | 0.944 |
| entropy | 9 | 1.000 | 0.944 |
| entropy | 10 | 1.000 | 0.944 |
|  | | | |
| log\_loss | 3 | 0.992 | 0.963 |
| log\_loss | 4 | 1.000 | 0.944 |
| log\_loss | 5 | 1.000 | 0.944 |
| log\_loss | 6 | 1.000 | 0.944 |
| log\_loss | 7 | 1.000 | 0.944 |
| log\_loss | 8 | 1.000 | 0.944 |
| log\_loss | 9 | 1.000 | 0.944 |
| log\_loss | 10 | 1.000 | 0.944 |



*Таблиця результатів методу випадкового лісу*

|  |  |  |  |
| --- | --- | --- | --- |
| Критерій розбиття | n\_estimators | Точність на навчальній вибірці | Точність на тестовій вибірці |
| gini | 2 | 0.935 | 0.870 |
| gini | 3 | 0.984 | 0.926 |
| gini | 4 | 1.000 | 0.963 |
| gini | 5 | 1.000 | 0.981 |
| gini | 6 | 1.000 | 1.000 |
| gini | 7 | 1.000 | 1.000 |
| gini | 8 | 1.000 | 0.981 |
| gini | 9 | 1.000 | 1.000 |
| gini | 10 | 1.000 | 1.000 |
| gini | 11 | 1.000 | 1.000 |
| gini | 12 | 1.000 | 1.000 |
| gini | 13 | 1.000 | 1.000 |
| gini | 14 | 1.000 | 1.000 |
| gini | 15 | 1.000 | 1.000 |
| gini | 16 | 1.000 | 1.000 |
| gini | 17 | 1.000 | 1.000 |
| gini | 18 | 1.000 | 1.000 |
| gini | 19 | 1.000 | 1.000 |
| gini | 20 | 1.000 | 1.000 |
|  | | | |
| entropy | 2 | 0.935 | 0.889 |
| entropy | 3 | 1.000 | 0.981 |
| entropy | 4 | 1.000 | 0.963 |
| entropy | 5 | 1.000 | 1.000 |
| entropy | 6 | 1.000 | 0.981 |
| entropy | 7 | 1.000 | 1.000 |
| entropy | 8 | 1.000 | 1.000 |
| entropy | 9 | 1.000 | 1.000 |
| entropy | 10 | 1.000 | 1.000 |
| entropy | 11 | 1.000 | 1.000 |
| entropy | 12 | 1.000 | 1.000 |
| entropy | 13 | 1.000 | 1.000 |
| entropy | 14 | 1.000 | 1.000 |
| entropy | 15 | 1.000 | 0.981 |
| entropy | 16 | 1.000 | 1.000 |
| entropy | 17 | 1.000 | 1.000 |
| entropy | 18 | 1.000 | 1.000 |
| entropy | 19 | 1.000 | 1.000 |
| entropy | 20 | 1.000 | 1.000 |
|  | | | |
| log\_loss | 2 | 0.935 | 0.889 |
| log\_loss | 3 | 1.000 | 0.981 |
| log\_loss | 4 | 1.000 | 0.963 |
| log\_loss | 5 | 1.000 | 1.000 |
| log\_loss | 6 | 1.000 | 0.981 |
| log\_loss | 7 | 1.000 | 1.000 |
| log\_loss | 8 | 1.000 | 1.000 |
| log\_loss | 9 | 1.000 | 1.000 |
| log\_loss | 10 | 1.000 | 1.000 |
| log\_loss | 11 | 1.000 | 1.000 |
| log\_loss | 12 | 1.000 | 1.000 |
| log\_loss | 13 | 1.000 | 1.000 |
| log\_loss | 14 | 1.000 | 1.000 |
| log\_loss | 15 | 1.000 | 0.981 |
| log\_loss | 16 | 1.000 | 1.000 |
| log\_loss | 17 | 1.000 | 1.000 |
| log\_loss | 18 | 1.000 | 1.000 |
| log\_loss | 19 | 1.000 | 1.000 |
| log\_loss | 20 | 1.000 | 1.000 |



**Код програми:**

**import** matplotlib.pyplot **as** plt

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.datasets **import** load\_wine

**from** sklearn.tree **import** DecisionTreeClassifier

**from** sklearn.ensemble **import** RandomForestClassifier

dataset = load\_wine()

X = dataset.data

y = dataset.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

train\_size=0.7, random\_state=0)

max\_depth\_values = range(3, 11)

criterion\_values = ['gini', 'entropy', 'log\_loss']

train\_scores = {criterion: [] **for** criterion **in** criterion\_values}

test\_scores = {criterion: [] **for** criterion **in** criterion\_values}

**for** criterion **in** criterion\_values:

*# Перебір різних значень глибини дерева*

**for** max\_depth **in** max\_depth\_values:

*# Створення моделі дерева рішень з поточними значеннями критерію та глибини*

        dtc = DecisionTreeClassifier(criterion=criterion, max\_depth=max\_depth, random\_state=0)

*# Навчання моделі на навчальній вибірці*

        dtc.fit(X\_train, y\_train)

        train\_accuracy = dtc.score(X\_train, y\_train)

        test\_accuracy = dtc.score(X\_test, y\_test)

        train\_scores[criterion].append(train\_accuracy)

        test\_scores[criterion].append(test\_accuracy)

*# Вивід інформації про поточну комбінацію гіперпараметрів та точність моделі на навчальній та тестовій вибірках*

        print(f'{criterion} | {max\_depth} | {train\_accuracy:.3f} | {test\_accuracy:.3f}')

print(train\_scores)

print(test\_scores)

**for** criterion **in** criterion\_values:

    plt.figure(figsize=(10, 6))

    plt.plot(max\_depth\_values, train\_scores[criterion], marker='o', label='Навчальна вибірка', color='deepskyblue')

    plt.plot(max\_depth\_values, test\_scores[criterion], marker='s', label='Тестова вибірка', color='darkorange')

    plt.xlabel('max\_depth')

    plt.ylabel('Точність')

    plt.title(f'Залежність точності від max\_depth (criterion={criterion})')

    plt.legend()

    plt.grid(True)

    plt.savefig(f"./plots/lab2\_decisiontree\_{criterion}")

    plt.show()

n\_estimator\_values = range(2, 21)

train\_scores\_rf = {criterion: [] **for** criterion **in** criterion\_values}

test\_scores\_rf = {criterion: [] **for** criterion **in** criterion\_values}

**for** criterion **in** criterion\_values:

**for** n\_estimator **in** n\_estimator\_values:

        rfc = RandomForestClassifier(n\_estimators=n\_estimator, criterion=criterion, random\_state=0)

        rfc.fit(X\_train, y\_train)

        train\_accuracy = rfc.score(X\_train, y\_train)

        test\_accuracy = rfc.score(X\_test, y\_test)

        train\_scores\_rf[criterion].append(train\_accuracy)

        test\_scores\_rf[criterion].append(test\_accuracy)

        print(f'{criterion} | {n\_estimator} | {train\_accuracy:.3f} | {test\_accuracy:.3f}')

print(train\_scores\_rf)

print(test\_scores\_rf)

**for** criterion **in** criterion\_values:

    plt.figure(figsize=(10, 6))

    plt.plot(n\_estimator\_values, train\_scores\_rf[criterion], marker='o', label='Навчальна вибірка')

    plt.plot(n\_estimator\_values, test\_scores\_rf[criterion], marker='s', label='Тестова вибірка')

    plt.xlabel('n\_estimators')

    plt.ylabel('Точність')

    plt.title(f'Залежність точності від n\_estimators (criterion={criterion})')

    plt.legend()

    plt.grid(True)

    plt.savefig(f"./plots/lab2\_randomforest\_{criterion}")

    plt.show()