

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

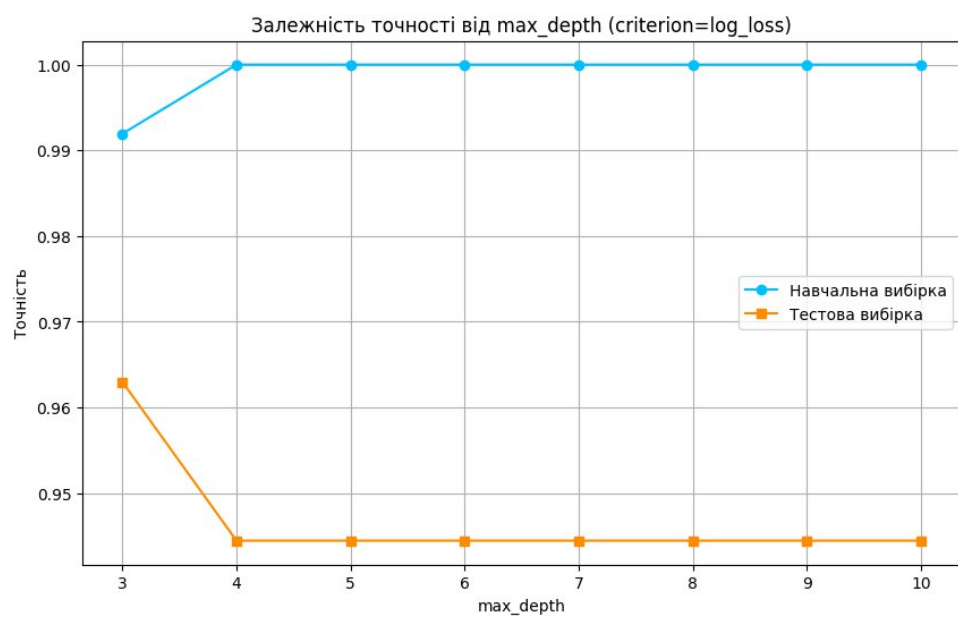
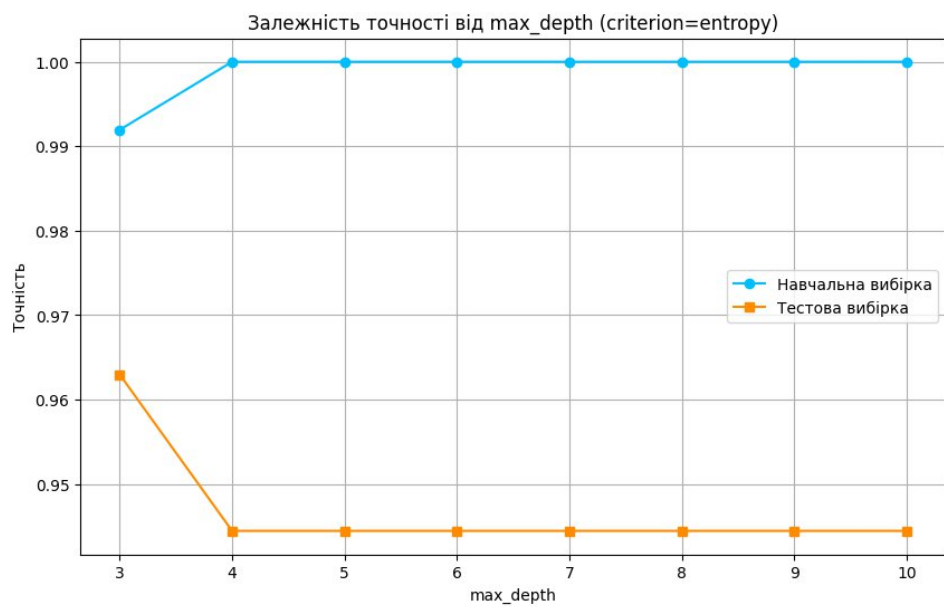
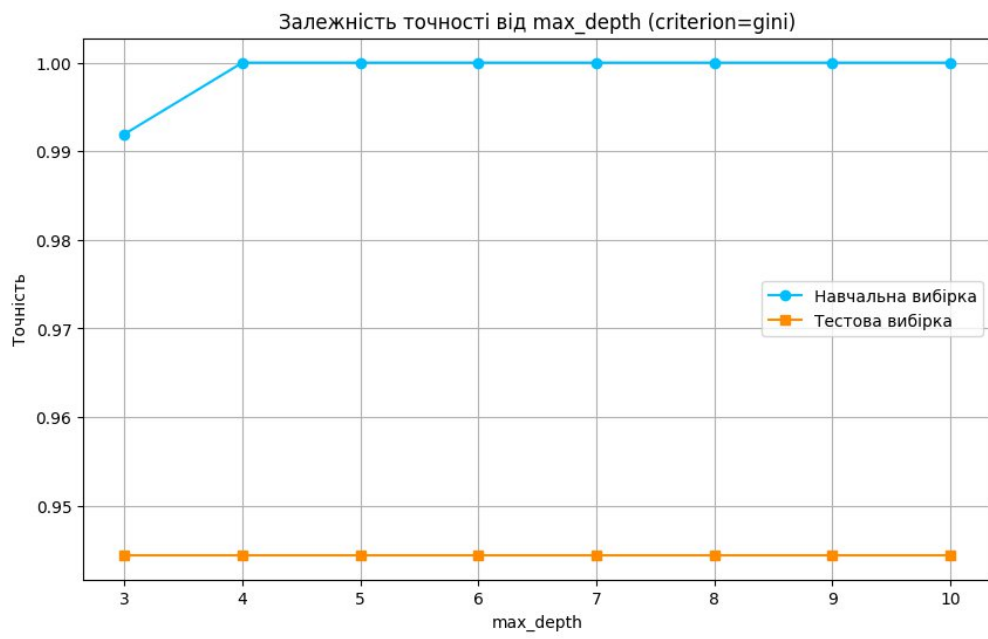
**Мета роботи:** вивчення застосувань дерев прийняття рішень та їх ансамблю для класифікації.

**Завдання на роботу:** завантаження набору даних, формування навчальної та тестової вибірок, визначення функціоналу якості класифікатора, візуалізація результатів.

Значення параметру `train_size = 0.7`

*Таблиця результатів дерева прийняття рішень*

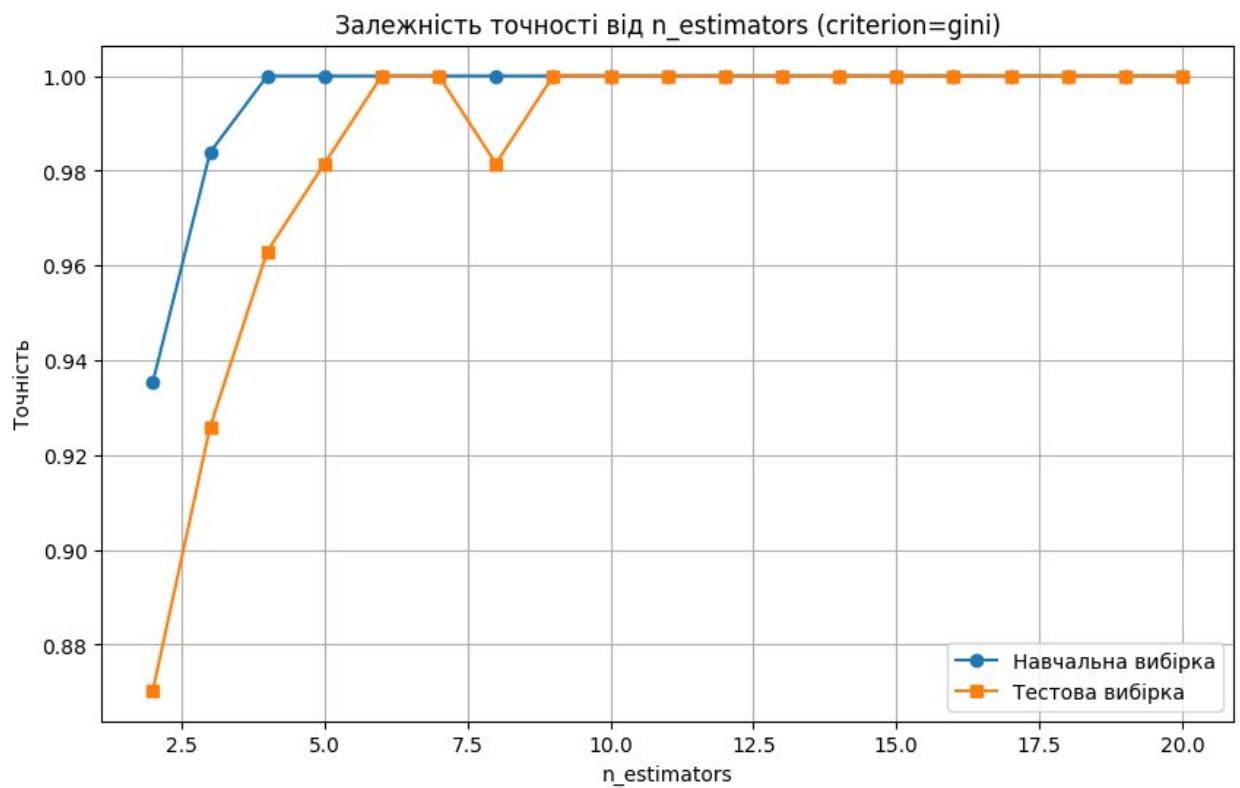
Критерій розбиття	max_depth	Точність на навчальній вибірці	Точність на тестовій вибірці
gini	3	0.992	0.944
gini	4	1.000	0.944
gini	5	1.000	0.944
gini	6	1.000	0.944
gini	7	1.000	0.944
gini	8	1.000	0.944
gini	9	1.000	0.944
gini	10	1.000	0.944
entropy	3	0.992	0.963
entropy	4	1.000	0.944
entropy	5	1.000	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

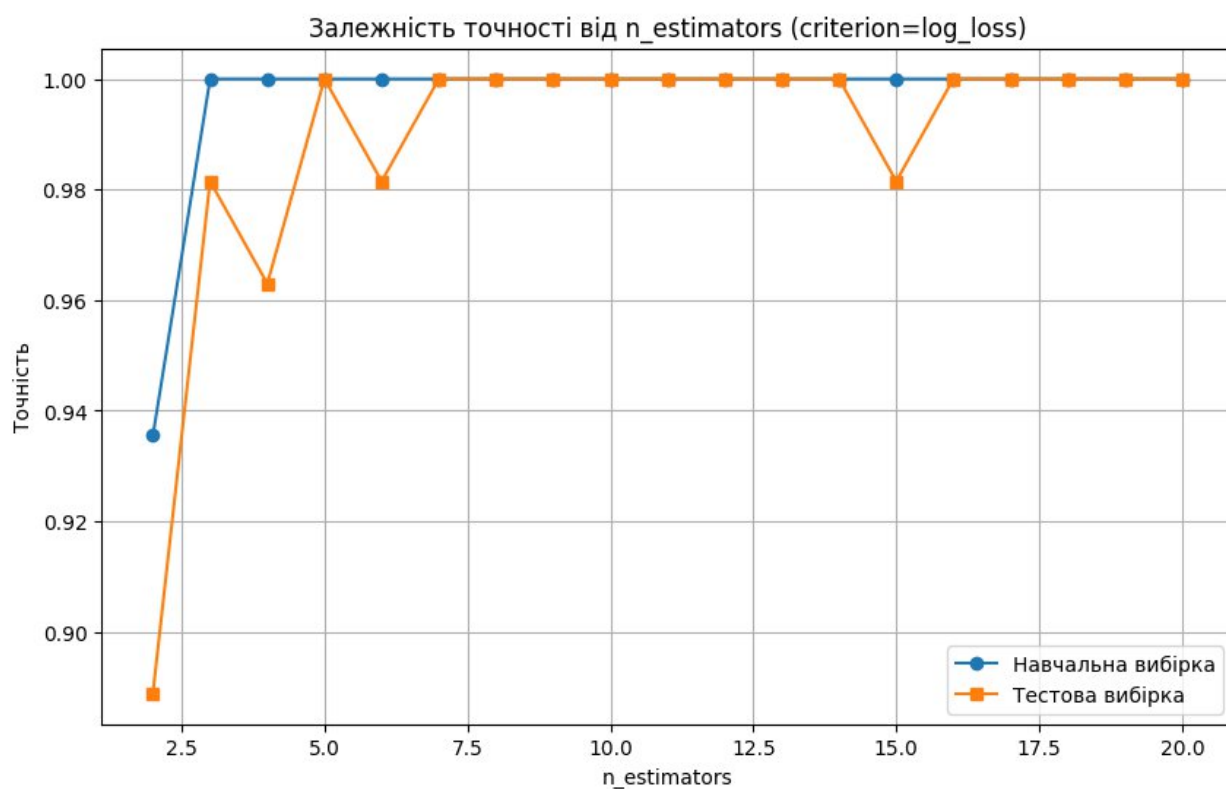
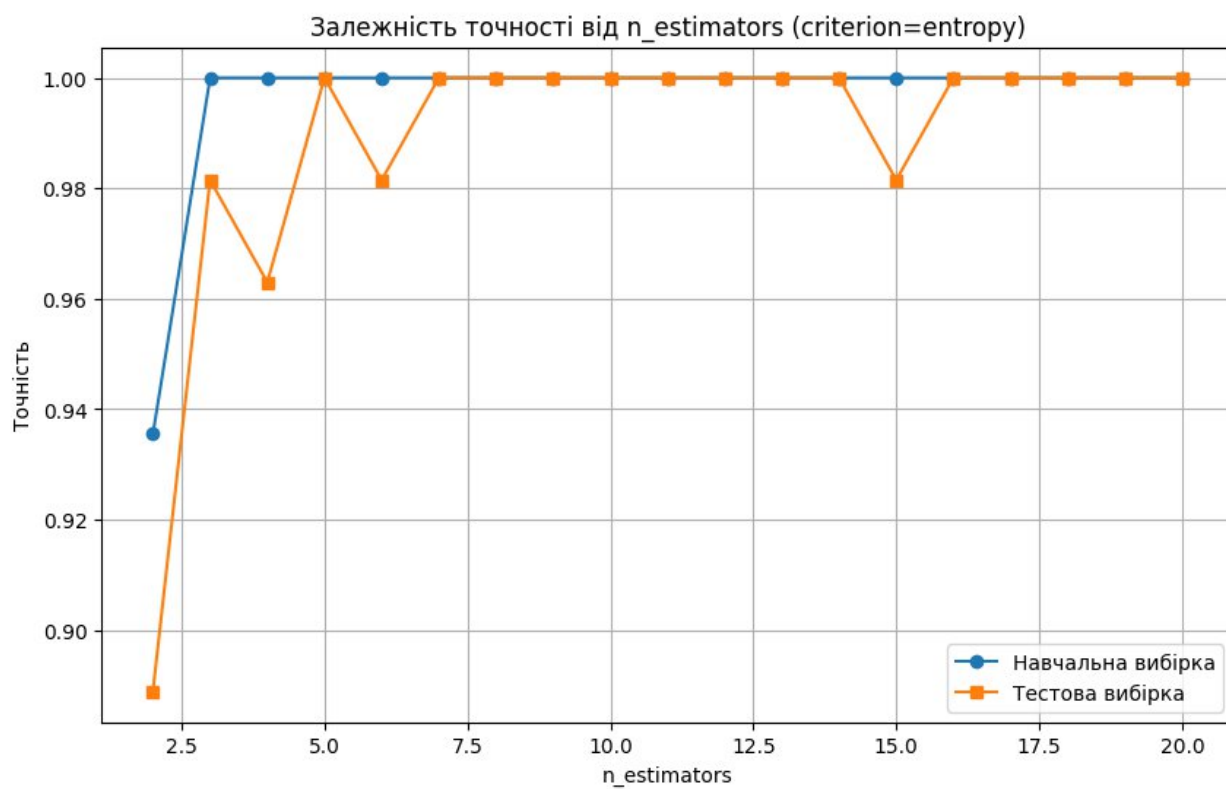


Таблиця результатів методу випадкового лісу

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_decisintree_{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()

```