

ЛАБОРАТОРНА РОБОТА № 5

Мета роботи: використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити методи ансамблів у машинному навчанні.

Завдання 1. Створення класифікаторів на основі випадкових та гранично випадкових лісів

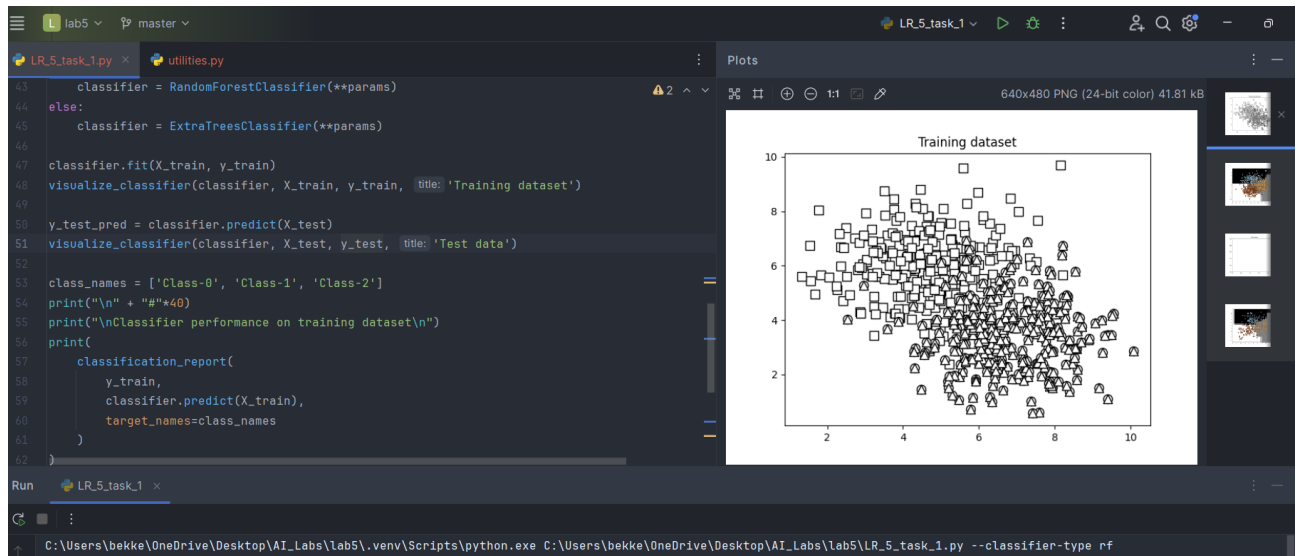


Рис 1. Візуалізація вхідних даних

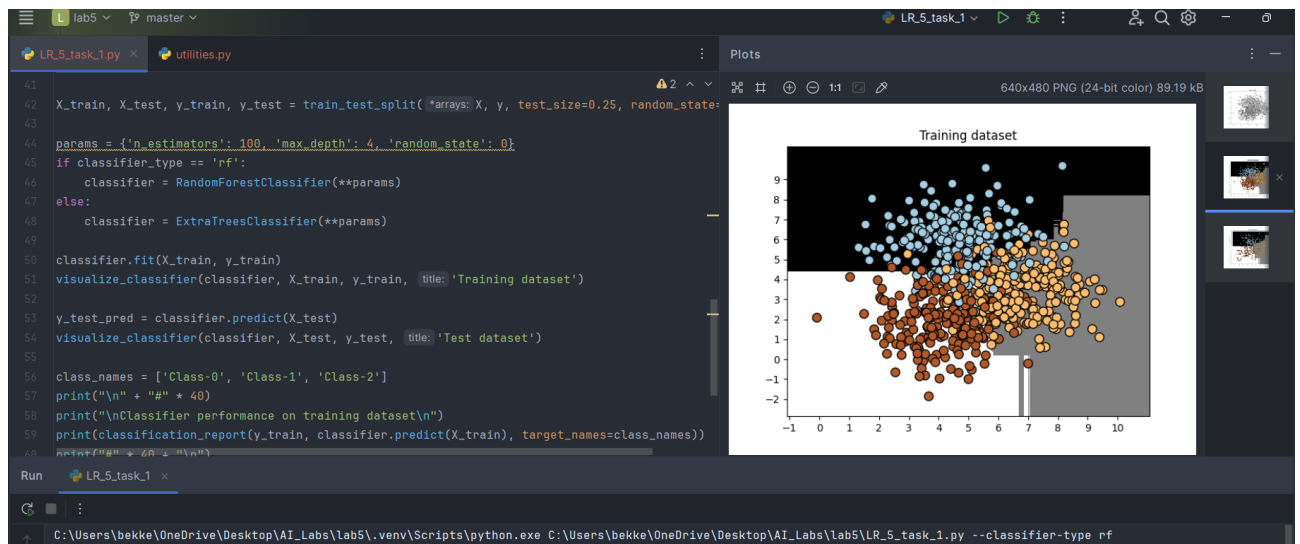


Рис 2. Візуалізація класифікації тренувальних даних (rf)

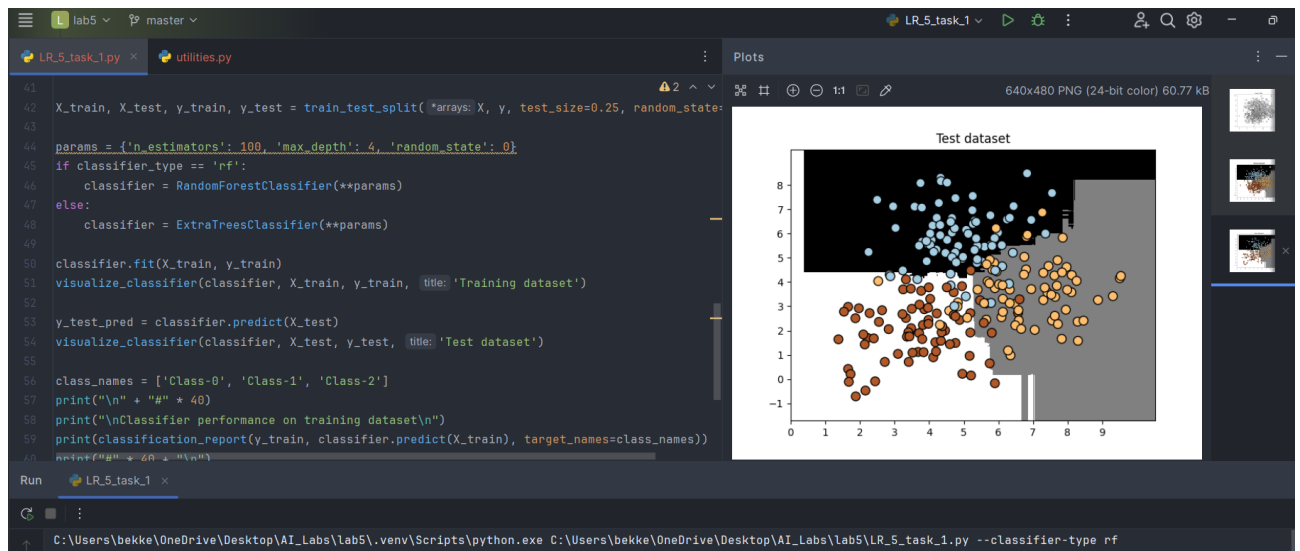


Рис 3. Візуалізація класифікації тестових даних (rf)

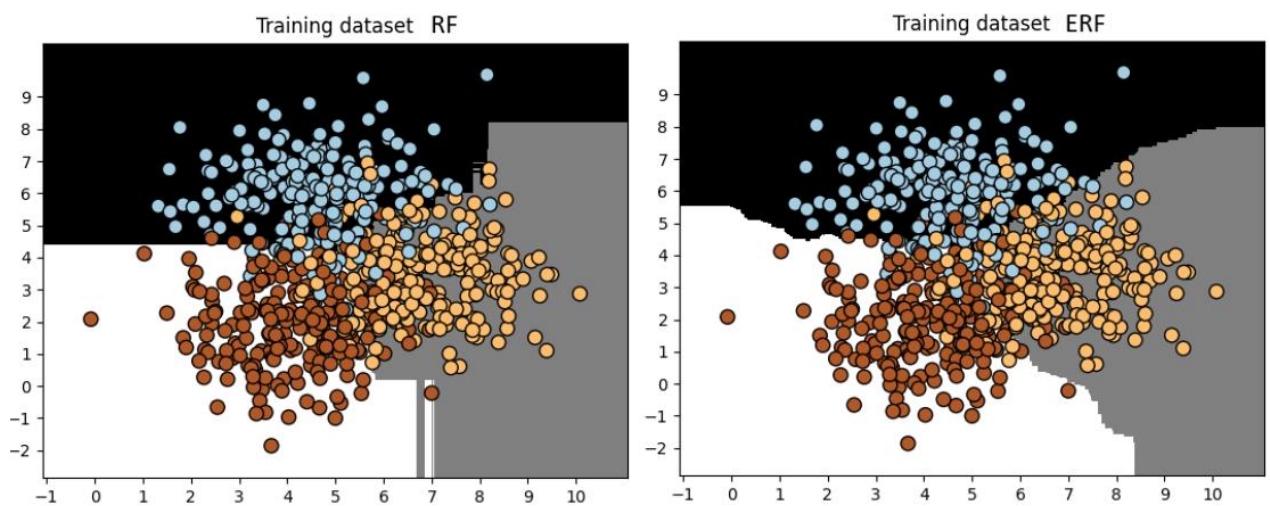


Рис 4. Порівняння графіків

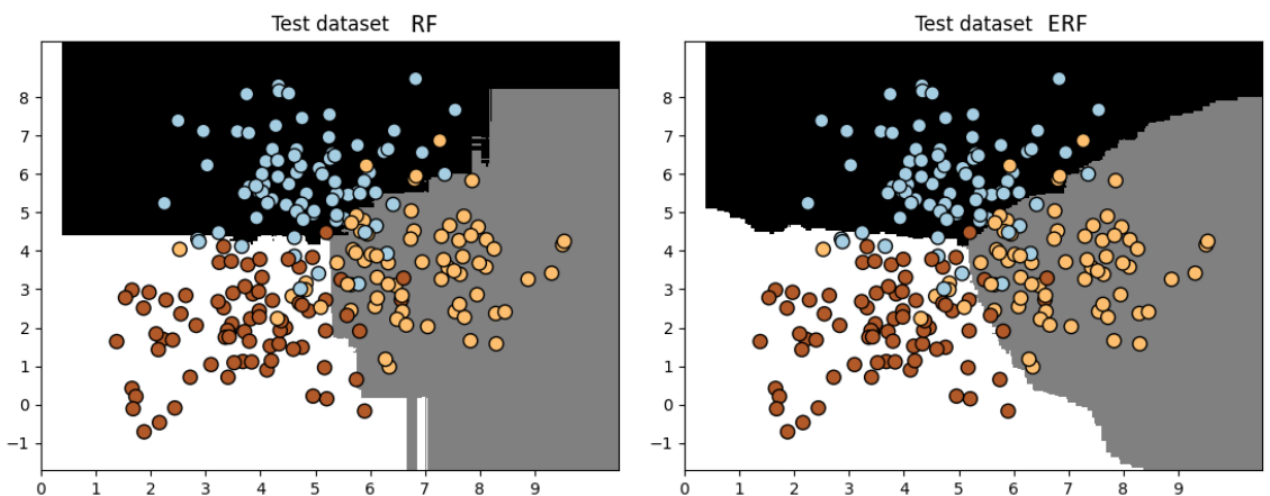


Рис 5. Порівняння графіків

Оцінка мір достовірності прогнозів

```
C:\Users\bekke\OneDrive\Desktop\AI_Labs\lab5\.venv\Scripts\python.exe C:\Users\bekke\OneDrive\Desktop\AI_Labs\lab5\LR_5_task_1.py --classifier-type rf

Confidence measure:

Datapoint: [5 5]
Predicted class: Class-0

Datapoint: [3 6]
Predicted class: Class-0

Datapoint: [6 4]
Predicted class: Class-1

Datapoint: [7 2]
Predicted class: Class-1

Datapoint: [4 4]
Predicted class: Class-2

Datapoint: [5 2]
Predicted class: Class-2

Process finished with exit code 0
```

Рис 6. Рівні довірливості (rf)

```
C:\Users\bekke\OneDrive\Desktop\AI_Labs\lab5\.venv\Scripts\python.exe C:\Users\bekke\OneDrive\Desktop\AI_Labs\lab5\LR_5_task_1.py --classifier-type erf

Confidence measure:

Datapoint: [5 5]
Predicted class: Class-0

Datapoint: [3 6]
Predicted class: Class-0

Datapoint: [6 4]
Predicted class: Class-1

Datapoint: [7 2]
Predicted class: Class-1

Datapoint: [4 4]
Predicted class: Class-2

Datapoint: [5 2]
Predicted class: Class-2

Process finished with exit code 0
```

Рис 7. Рівні довірливості (erf)

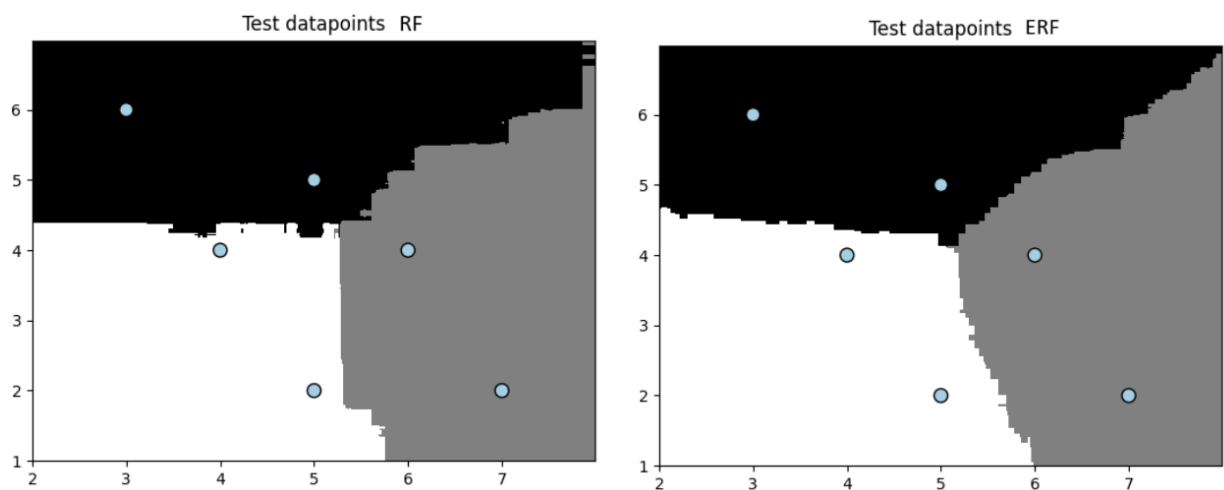


Рис 8. Графіки функцій

Classifier performance on training dataset					Classifier performance on test dataset				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Class-0	0.91	0.86	0.88	221	Class-0	0.92	0.85	0.88	79
Class-1	0.84	0.87	0.86	230	Class-1	0.86	0.84	0.85	70
Class-2	0.86	0.87	0.86	224	Class-2	0.84	0.92	0.88	76
accuracy			0.87	675	accuracy			0.87	225
macro avg	0.87	0.87	0.87	675	macro avg	0.87	0.87	0.87	225
weighted avg	0.87	0.87	0.87	675	weighted avg	0.87	0.87	0.87	225

Рис 9. Оцінка якості (RF)

Classifier performance on training dataset					Classifier performance on test dataset				
	precision	recall	f1-score	support		precision	recall	f1-score	support
Class-0	0.89	0.83	0.86	221	Class-0	0.92	0.85	0.88	79
Class-1	0.82	0.84	0.83	230	Class-1	0.84	0.84	0.84	70
Class-2	0.83	0.86	0.85	224	Class-2	0.85	0.92	0.89	76
accuracy			0.85	675	accuracy			0.87	225
macro avg	0.85	0.85	0.85	675	macro avg	0.87	0.87	0.87	225
weighted avg	0.85	0.85	0.85	675	weighted avg	0.87	0.87	0.87	225

Рис 10. Оцінка якості (ERF)

Висновок: на даному етапі вдалося порівняти два окремих випадка ансамблевого навчання: «Випадковий ліс» та «Гранично випадковий ліс», за допомогою візуалізації було отримано результати, на яких видно, що дійсно, «Гранично випадковий ліс» призводить до більш гладких меж прийняття рішень, що означає меншу варіативність моделі. Також було отримано результати оцінки якості, що виявились близьким за значенням для обох випадків ансамблевого навчання.

Завдання 2. Обробка дисбалансу класів

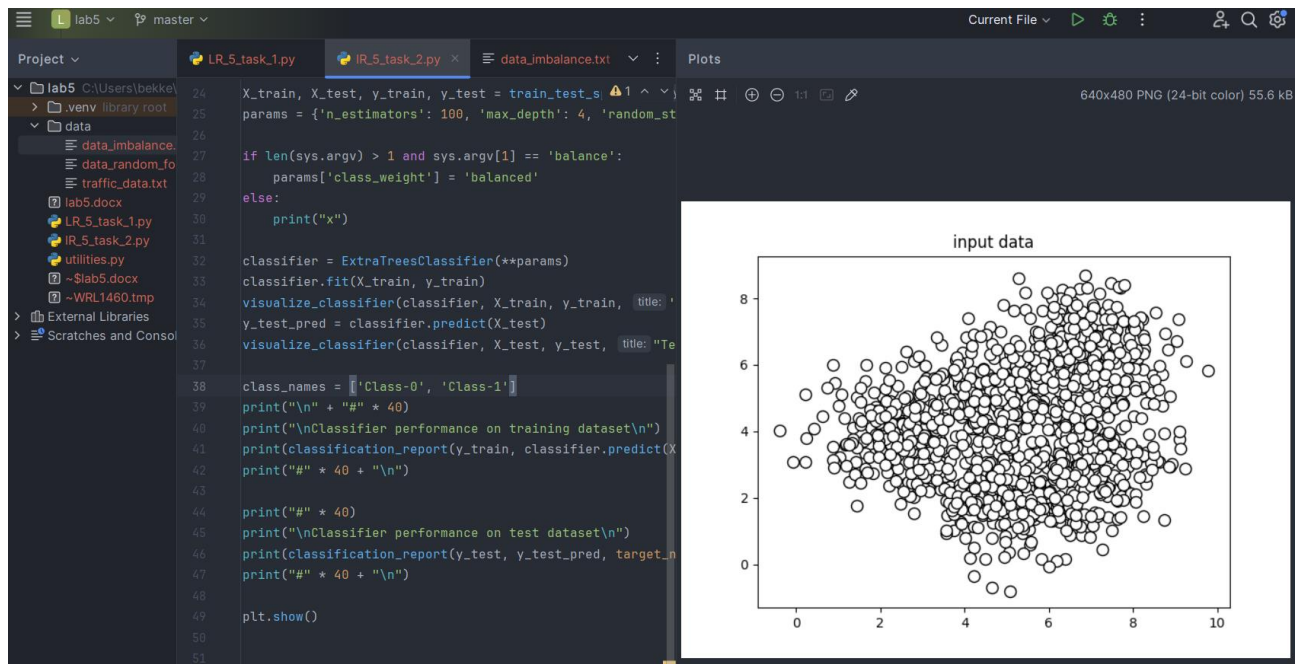


Рис 11. Вхідні дані

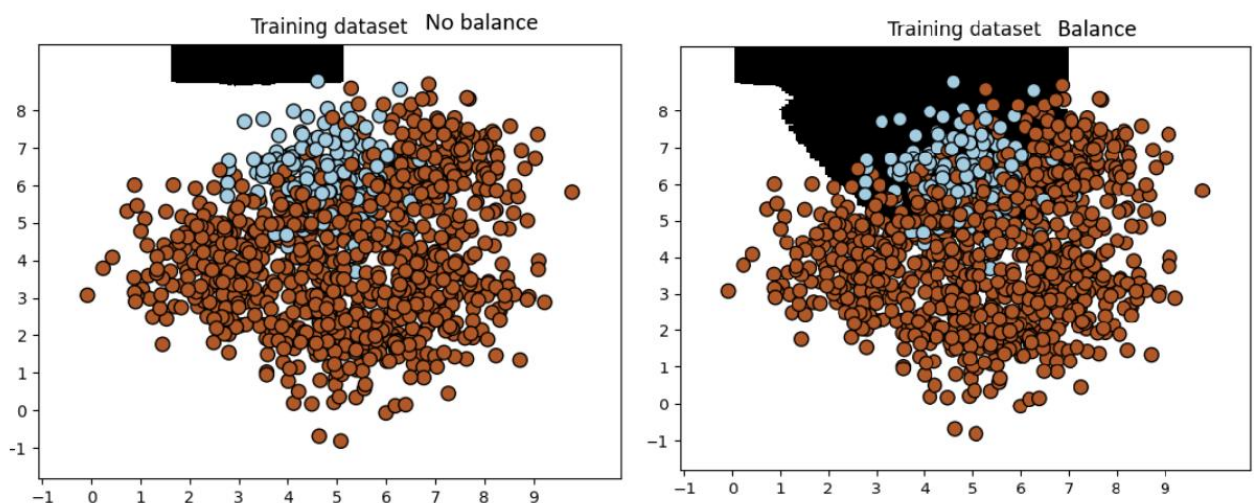


Рис 12. Графік даних класифікатора для тестового набору

```

C:\Users\bekke\OneDrive\Desktop\AI_Labs\lab5\.venv\Scripts\python.exe C:\Users\bekke\OneDrive\Desktop\AI_Labs\lab5\lr_5_task_2.py balance
C:\Users\bekke\OneDrive\Desktop\AI_Labs\lab5\lr_5_task_2.py:18: UserWarning: You passed a edgecolor/edgecolors ('black') for an unfilled marker
  plt.scatter(class_0[:, 0], class_0[:, 1], s=75, facecolors='white',

#####

Classifier performance on training dataset

      precision    recall  f1-score   support

   Class-0       0.44      0.93      0.60       181
   Class-1       0.98      0.77      0.86       944

 accuracy          0.80          1125
 macro avg       0.71      0.85      0.73       1125
weighted avg       0.89      0.80      0.82       1125

#####

#####

Classifier performance on test dataset

      precision    recall  f1-score   support

   Class-0       0.45      0.94      0.61        69
   Class-1       0.98      0.74      0.84       306

 accuracy          0.78          375
 macro avg       0.72      0.84      0.73       375
weighted avg       0.88      0.78      0.80       375

#####

```

Рис 13. Оцінка якості (with balance)

Висновок: за допомогою візуалізації та оцінки якості було доведено, що незбалансовані дані погано впливають на якість роботи класифікатора. Для незбалансованих даних також не вдалося визначити фактичну межу між двома класами (див: рис 13).

Завдання 3. Знаходження оптимальних навчальних параметрів за допомогою сіткового пошуку


```
7 from utilities import visualize_classifier
8
9 input_file = 'data/data_random_forests.txt'
10 data = np.loadtxt(input_file, delimiter=',')
11 X, y = data[:, :-1], data[:, -1]
12
13
14 class_0 = np.array(X[y == 0])
15 class_1 = np.array(X[y == 1])
16 class_2 = np.array(X[y == 2])
17
18 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
19
20 parameter_grid = [{'n_estimators': [100], 'max_depth': [2, 4, 7, 12, 16]},
21                  {'max_depth': [4], 'n_estimators': [25, 50, 100, 250]}]
22
23 metrics = ['recall_weighted', 'precision_weighted']
24
25 classifier = None
26 for metric in metrics:
27     print("\n#### Searching optimal parameters for", metric)
28     classifier = GridSearchCV(
29         ExtraTreesClassifier(random_state=0),
30         parameter_grid, cv=5, scoring=metric
31     )
32     classifier.fit(X_train, y_train)
33     print("\nGrid scores for the parameter grid:")
34     for params, avg_score in classifier.cv_results_.items():
35         print(params, '-->', avg_score, 3)
36
37     print("\nBest parameters:", classifier.best_params_)
38     y_pred = classifier.predict(X_test)
```

Рис 14. Результати пошуку оптимальних параметрів для recall

```
16 class_2 = np.array(X[y == 2])
17
18 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
19
20 parameter_grid = [{'n_estimators': [100], 'max_depth': [2, 4, 7, 12, 16]},
21                  {'max_depth': [4], 'n_estimators': [25, 50, 100, 250]}]
22
23 metrics = ['recall_weighted', 'precision_weighted']
24
25 classifier = None
26 for metric in metrics:
27     print("\n#### Searching optimal parameters for", metric)
28     classifier = GridSearchCV(
29         ExtraTreesClassifier(random_state=0),
30         parameter_grid, cv=5, scoring=metric
31     )
32     classifier.fit(X_train, y_train)
33     print("\nGrid scores for the parameter grid:")
34     for params, avg_score in classifier.cv_results_.items():
35         print(params, '-->', avg_score, 3)
36
37     print("\nBest parameters:", classifier.best_params_)
38     y_pred = classifier.predict(X_test)
39     print("\nPerformance report:\n")
40     print(classification_report(y_test, y_pred))
41
42
```

Рис 15. Результати пошуку оптимальних параметрів для precision

Best params & performance for recall

Best parameters: {'max_depth': 2, 'n_estimators': 100}

Performance report:

	precision	recall	f1-score	support	
	0.0	0.94	0.81	0.87	79
	1.0	0.81	0.86	0.83	70
	2.0	0.83	0.91	0.87	76
accuracy			0.86	225	
macro avg	0.86	0.86	0.86	225	
weighted avg	0.86	0.86	0.86	225	

Best params & performance for precision

Best parameters: {'max_depth': 2, 'n_estimators': 100}

Performance report:

	precision	recall	f1-score	support	
	0.0	0.94	0.81	0.87	79
	1.0	0.81	0.86	0.83	70
	2.0	0.83	0.91	0.87	76
accuracy			0.86	225	
macro avg	0.86	0.86	0.86	225	
weighted avg	0.86	0.86	0.86	225	

Рис 16. Найкращі параметри та оцінка продуктивності

Висновок: на даному етапі вдалося виявити, що найкращими оптимізуючими параметрами для precision & recall є $\text{max_depth} = 2$ & $\text{n_estimatirs} = 100$.
Відповідно до оцінок продуктивності: найвищі показники має клас 0, тобто класифікатор найкраще його розпізнає.

Завдання 4. Обчислення відносної важливості ознак

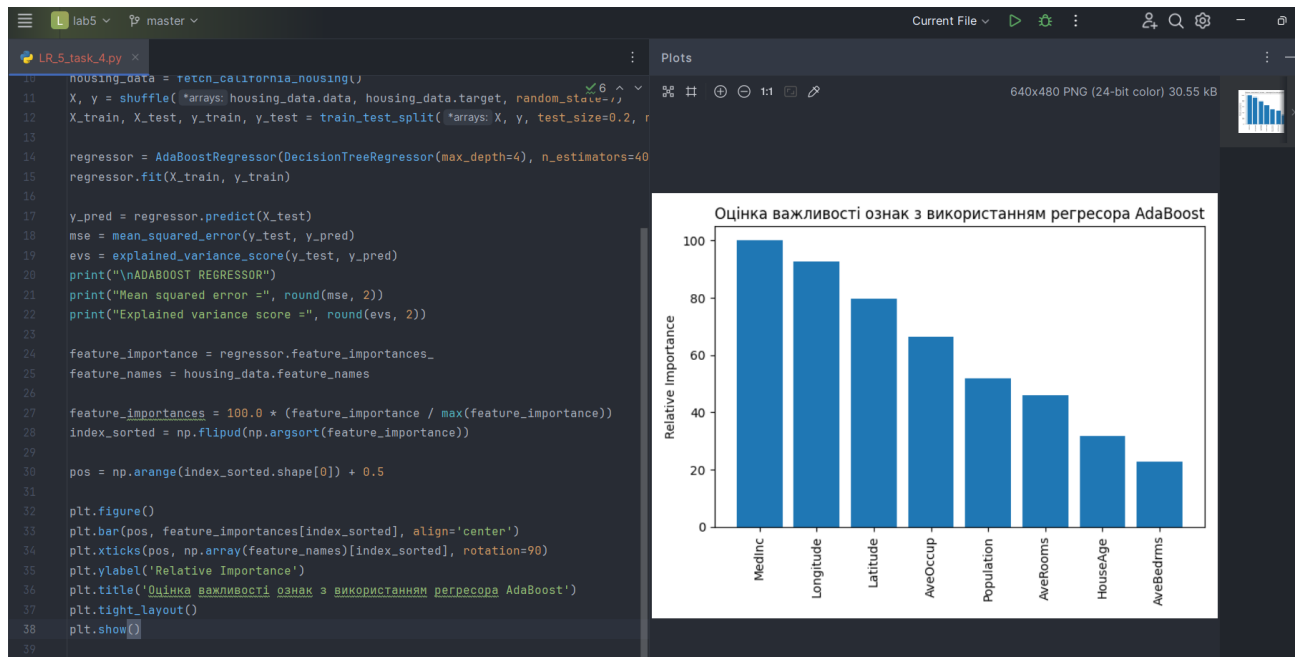


Рис 17. Графік

```
C:\Users\bekke\OneDrive\Desktop\AI_Labs\lab5\.venv\Scripts\python.exe

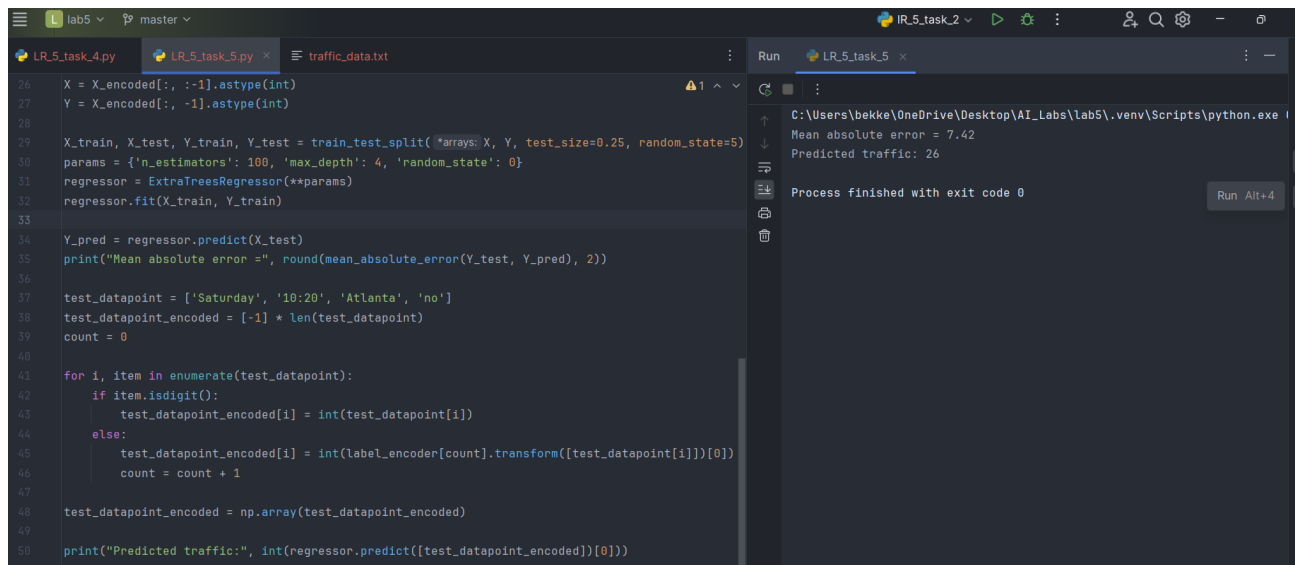
ADABOOST REGRESSOR
Mean squared error = 1.18
Explained variance score = 0.47

Process finished with exit code 0
```

Рис 18. Метрики

Висновок: в результаті аналізу було виявлено, що найбільшу роль мають ознаки: medinc, longitude, latitude, тоді як останніми двома-трьома можна знехтувати. Відповідно до метрик: середньоквадратична помилка 1.18, якщо дані це просто цифри, є низькою, тобто відхилення не дуже велике, що є гарними результатами. Дисперсія = 0.47 є середнім результатом, що свідчить про те, що модель здатна пояснювати майже половину варіацій у даних.

Завдання 5. Прогнозування інтенсивності дорожнього руху за допомогою класифікатора на основі гранично випадкових лісів



```
26 X = X_encoded[:, :-1].astype(int)
27 Y = X_encoded[:, -1].astype(int)
28
29 X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=5)
30 params = {'n_estimators': 100, 'max_depth': 4, 'random_state': 0}
31 regressor = ExtraTreesRegressor(**params)
32 regressor.fit(X_train, Y_train)
33
34 Y_pred = regressor.predict(X_test)
35 print("Mean absolute error =", round(mean_absolute_error(Y_test, Y_pred), 2))
36
37 test_datapoint = ['Saturday', '10:20', 'Atlanta', 'no']
38 test_datapoint_encoded = [-1] * len(test_datapoint)
39 count = 0
40
41 for i, item in enumerate(test_datapoint):
42     if item.isdigit():
43         test_datapoint_encoded[i] = int(test_datapoint[i])
44     else:
45         test_datapoint_encoded[i] = int(label_encoder[count].transform([test_datapoint[i]]))
46         count = count + 1
47
48 test_datapoint_encoded = np.array(test_datapoint_encoded)
49
50 print("Predicted traffic:", int(regressor.predict([test_datapoint_encoded])[0]))
```

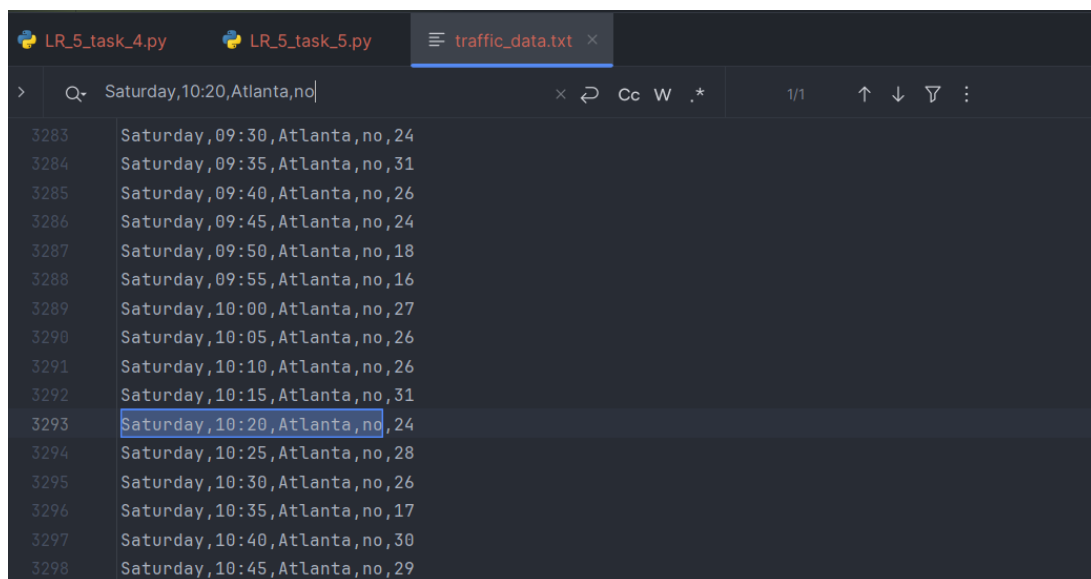
Run LR_5_task_5

C:\Users\bekke\OneDrive\Desktop\AI_Labs\lab5\.venv\Scripts\python.exe

Mean absolute error = 7.42
Predicted traffic: 26

Process finished with exit code 0

Рис 19. Метрики і прогноз



>	Q	Saturday,10:20,Atlanta,no	× ↺ Cc W .*	1/1	↑ ↓ 🔍 ⋮
3283		Saturday,09:30,Atlanta,no,24			
3284		Saturday,09:35,Atlanta,no,31			
3285		Saturday,09:40,Atlanta,no,26			
3286		Saturday,09:45,Atlanta,no,24			
3287		Saturday,09:50,Atlanta,no,18			
3288		Saturday,09:55,Atlanta,no,16			
3289		Saturday,10:00,Atlanta,no,27			
3290		Saturday,10:05,Atlanta,no,26			
3291		Saturday,10:10,Atlanta,no,26			
3292		Saturday,10:15,Atlanta,no,31			
3293		Saturday,10:20,Atlanta,no,24			
3294		Saturday,10:25,Atlanta,no,28			
3295		Saturday,10:30,Atlanta,no,26			
3296		Saturday,10:35,Atlanta,no,17			
3297		Saturday,10:40,Atlanta,no,30			
3298		Saturday,10:45,Atlanta,no,29			

Рис 20. Актуальні дані

Оскільки прогноз дійсно близький до фактичного значення, можна зробити висновок, що ансамблеве навчання дійсно має гарні показники на таких наборах даних.

Висновок: на даній лабораторній роботі, я, за допомогою спеціалізованих бібліотек та мови програмування, навчився створювати та аналізувати класифікатори на основі випадкових та гранично випадкових лісів. Також отримав практичні навички з обробки дисбалансу класів, навчився знаходити

оптимальні параметри за допомогою сіткового пошуку та обчислювати відносну важливість ознак.

Github: [link](#)