

ЛАБОРАТОРНА РОБОТА №2

ПОРІВНЯННЯ МЕТОДІВ КЛАСИФІКАЦІЇ ДАНИХ

Мета заняття: використовуючи спеціалізовані бібліотеки та мову програмування Python дослідити різні методи класифікації даних та навчитися їх порівнювати.

Хід роботи

Завдання 2.1. Класифікація за допомогою машин опорних векторів (SVM)

Назва	Опис	Тип значень
age	Вік	Числове
workclass	Вид працевлаштування	Категоріальне
fnlwtg	Кількість осіб з такими ж ознаками	Числове
education	Навчання	Категоріальне
education-num	Років навчання	Числове
marital-status	Сімейне положення	Категоріальне
occupation	Професія	Категоріальне
relationship	Відносини	Категоріальне
race	Раса	Категоріальне
sex	Стать	Категоріальне
capital-gain	Приріст капіталу	Числове
capital-loss	Втрата капіталу	Числове
hours-per-week	Кількість робочих годин на тиждень	Числове
native-country	Країна походження	Категоріальне

```
import numpy as np
from sklearn import preprocessing
from sklearn.svm import LinearSVC
from sklearn.multiclass import OneVsOneClassifier
```

ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.05.000 – Лр2

Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Горелко О. В.			Звіт з лабораторної роботи	Лім.	Арк.
Перевір.		Пулеко І. В.					Аркушів
Керівник							1
Н. контр.							3
Зав. каф.						ФІКТ Гр. ПІ-59	

```

from sklearn.model_selection import train_test_split, cross_val_score

# Вхідний файл, який містить дані
input_file = 'income_data.txt'

# Читання даних
X = []
y = []
count_class1 = 0
count_class2 = 0
max_datapoints = 25000

with open(input_file, 'r') as f:
    for line in f.readlines():
        if count_class1 >= max_datapoints and count_class2 >= max_datapoints:
            break

        if '?' in line:
            continue

        data = line[:-1].split(' ')

        if data[-1] == '<=50K' and count_class1 < max_datapoints:
            X.append(data)
            count_class1 += 1

        if data[-1] == '>50K' and count_class2 < max_datapoints:
            X.append(data)
            count_class2 += 1

# Перетворення на масив numpy
X = np.array(X)

# Перетворення рядкових даних на числові
label_encoder = []
X_encoded = np.empty(X.shape)
for i, item in enumerate(X[0]):
    if item.isdigit():
        X_encoded[:, i] = X[:, i]
    else:
        current_label_encoder = preprocessing.LabelEncoder()
        label_encoder.append(current_label_encoder)
        X_encoded[:, i] = current_label_encoder.fit_transform(X[:, i])

X = X_encoded[:, :-1].astype(int)
y = X_encoded[:, -1].astype(int)

classifier = OneVsOneClassifier(LinearSVC(random_state=0))
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)
classifier.fit(X_train, y_train)
y_test_pred = classifier.predict(X_test)

accuracy = cross_val_score(classifier, X, y, scoring='accuracy', cv=3)
print("Accuracy score: " + str(round(100 * accuracy.mean(), 2)) + "%")

precision = cross_val_score(classifier, X, y, scoring='precision', cv=3)
print("Precision score: " + str(round(100 * precision.mean(), 2)) + "%")

recall = cross_val_score(classifier, X, y, scoring='recall', cv=3)
print("Recall score: " + str(round(100 * recall.mean(), 2)) + "%")

f1 = cross_val_score(classifier, X, y, scoring='f1_weighted', cv=3)

```

		Горелко О. В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.05.000 – Лр2	Арк.
		Пудеко І. В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

print("F1 score: " + str(round(100 * f1.mean(), 2)) + "%")

# Передбачення результату для тестової точки даних
input_data = ['37', 'Private', '215646', 'HS-grad', '9', 'Never-married', 'Hand-
dlers-cleaners', 'Not-in-family', 'White',
              'Male', '0', '0', '40', 'United-States']

# Кодування тестової точки даних
input_data_encoded = [-1] * len(input_data)
count = 0
for i, item in enumerate(input_data):
    if item.isdigit():
        input_data_encoded[i] = int(input_data[i])
    else:
        input_data_encoded[i] = int(label_encoder[count].transform([(in-
put_data[i])])[-1])
        count += 1

input_data_encoded = np.array(input_data_encoded)

# Використання класифікатора для кодованої точки даних
# та виведення результату
predicted_class = classifier.predict([input_data_encoded])
print(label_encoder[-1].inverse_transform(predicted_class)[0])

```

Рисунок 1. Код програми

Accuracy score: 62.64%

Precision score: 69.18%

Recall score: 38.24%

F1 score: 56.15%

Тестова точка - <=50K. Отже тестова точка має дохід менше 50 тисяч в рік.

Завдання 2.2. Порівняння якості класифікаторів SVM з нелінійними ядрами

Поліноміальне ядро

Accuracy score: 58.41%

Precision score: 41.6%

Recall score: 33.05%

F1 score: 46.5%

>50K

Гаусове ядро

Accuracy score: 78.61%

Precision score: 98.72%

Recall score: 14.26%

F1 score: 71.95%

>50K

		Горелко О. В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.05.000 – Лр2	Арк.
		Пудеко І. В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

		Горелко О. В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.05.000 – Лр2	Арк.
		Пудеко І. В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

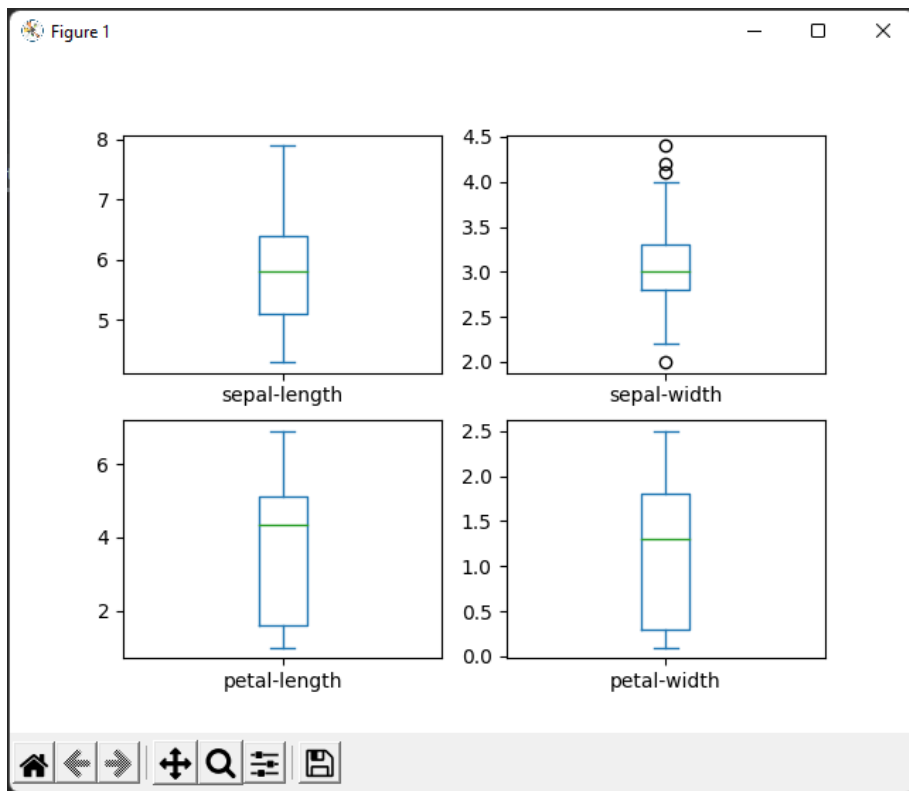


Рисунок 2. Одновимірні графіки

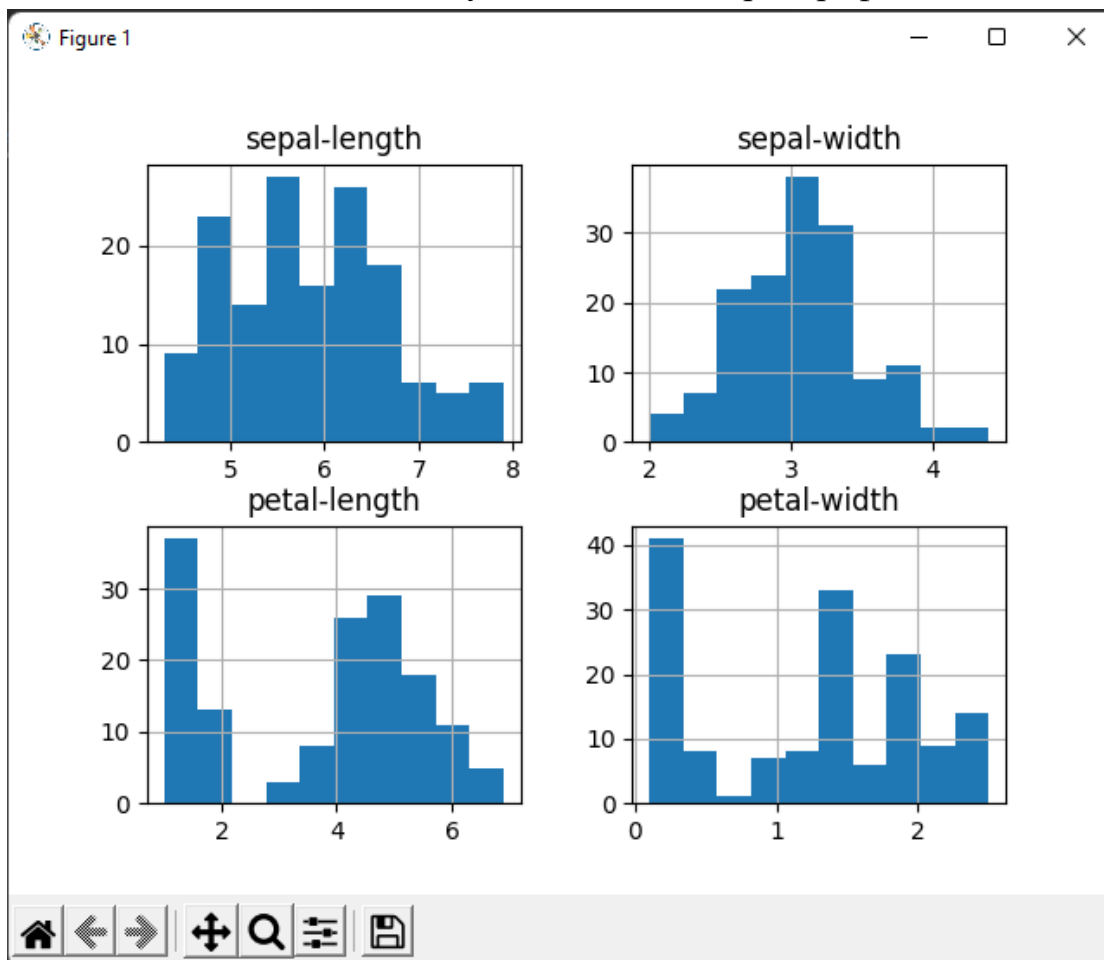


Рисунок 3. Діаграма розмаху атрибутів вхідних даних

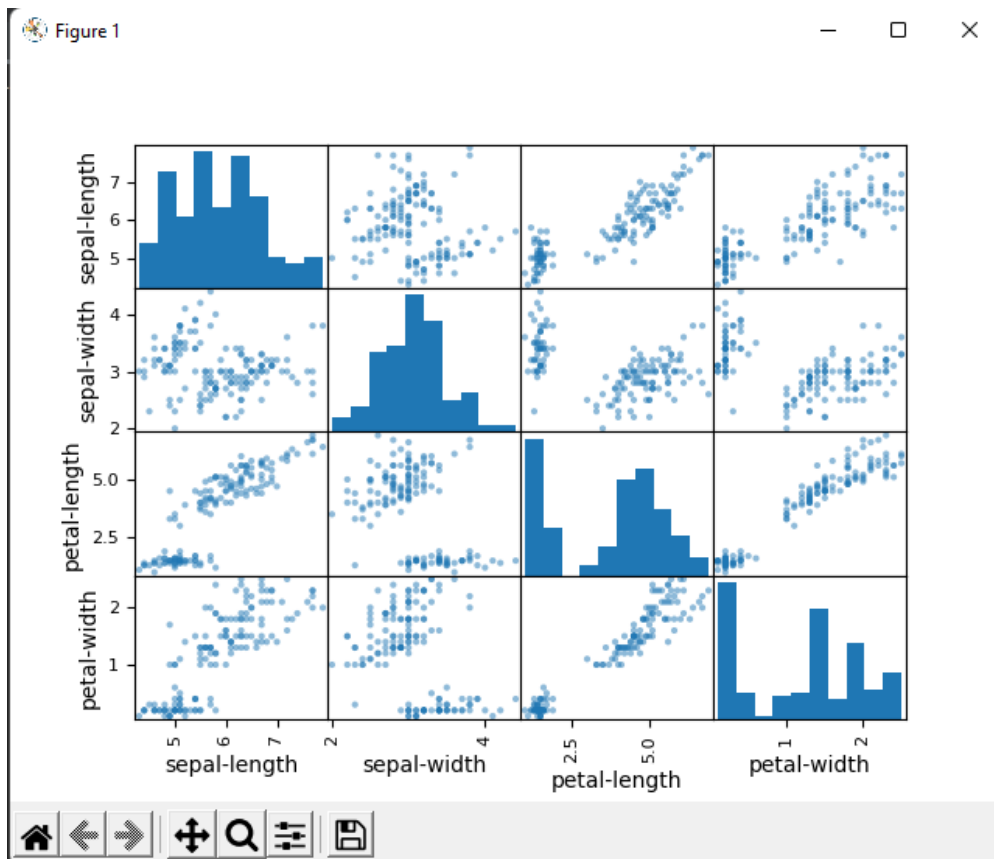


Рисунок 4. Матриця діаграм розсіювання

```
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"
names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
dataset = read_csv(url, names=names)

dataset.plot(kind='box', subplots=True, layout=(2, 2), sharex=False, sharey=False)
pyplot.show()

dataset.hist()
pyplot.show()

scatter_matrix(dataset)
pyplot.show()
```

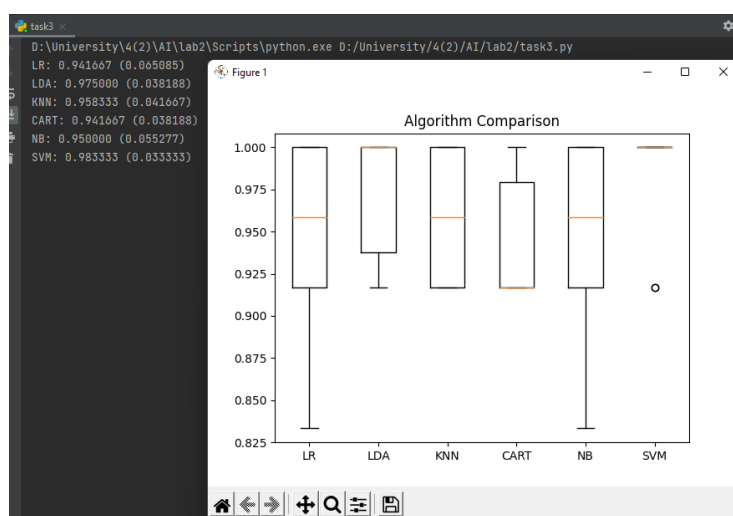


Рисунок 5. Порівняння алгоритмів

		Горелко О. В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.05.000 – Лр2	Арк.
		Пудеко І. В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

Проаналізувавши оптимальний графік, я обрав метод класифікації SVM, тому що він показав найвищу якість.

```

44 # оцінюємо модель на кожній ітерації
45 results = []
46 names = []
47 for name, model in models:
48     kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
49     cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')
50     results.append(cv_results)
51     names.append(name)
52     print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
53
54 pyplot.boxplot(results, labels=names)
55 pyplot.title('Algorithm Comparison')
56 pyplot.show()
57
58 model = SVC(gamma='auto')
59 model.fit(X_train, Y_train)
60 predictions = model.predict(X_validation)
61
62 print(accuracy_score(Y_validation, predictions))
63 print(confusion_matrix(Y_validation, predictions))
64 print(classification_report(Y_validation, predictions))
65
66 X_new = np.array([[5, 2.9, 1, 0.2]])
67 print("форма массива X_new: {}".format(X_new.shape))
68
69 prediction = model.predict(X_new)
70 print("Прогноз: {}".format(prediction))
71 print("Спрогнозована мітка: {}".format(prediction[0]))

```

Run: task3

```

D:\University\4(2)\AI\lab2\Scripts\python.exe D:/University/4(2)/AI/lab2/task3.py
LR: 0.941667 (0.065085)
LDA: 0.975000 (0.038188)
KNN: 0.958333 (0.041667)
CART: 0.941667 (0.053359)
NB: 0.950000 (0.055277)
SVM: 0.983333 (0.033333)
0.9666666666666667
[[11  0  0]
 [ 0 12  1]
 [ 0  0  6]]

```

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	11
Iris-versicolor	1.00	0.92	0.96	13
Iris-virginica	0.86	1.00	0.92	6
accuracy			0.97	30
macro avg	0.95	0.97	0.96	30
weighted avg	0.97	0.97	0.97	30

```

форма массива X_new: (1, 4)
Прогноз: ['Iris-setosa']
Спрогнозована мітка: Iris-setosa

Process finished with exit code 0

```

Рисунок 6. Результат

		Горелко О. В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.05.000 – Лр2	Арк.
		Пудеко І. В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.4. Порівняння якості класифікаторів для набору даних завдання 2.1

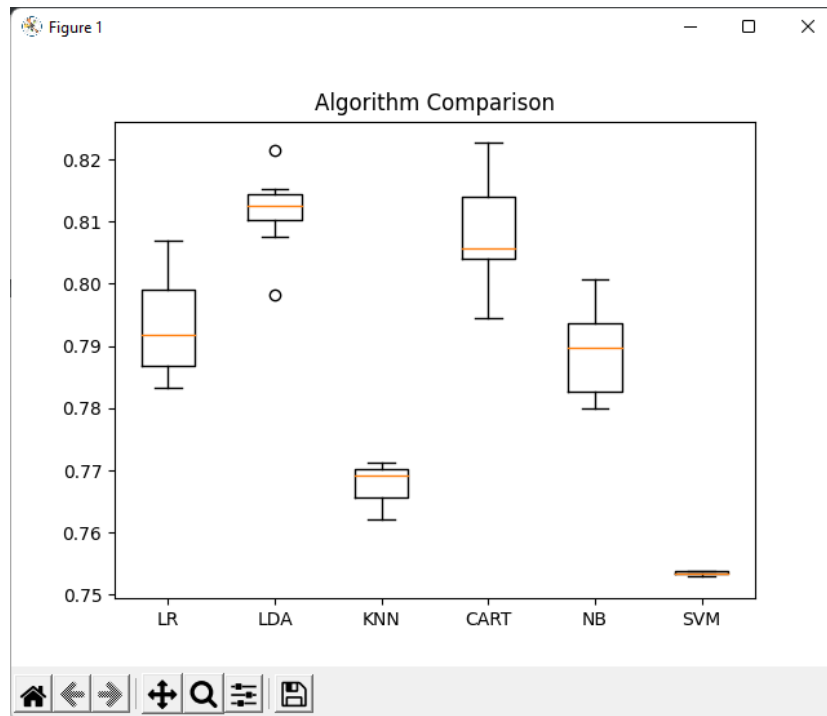


Рисунок 7. Порівняння алгоритмів

Завдання 2.5. Класифікація даних лінійним класифікатором Ridge

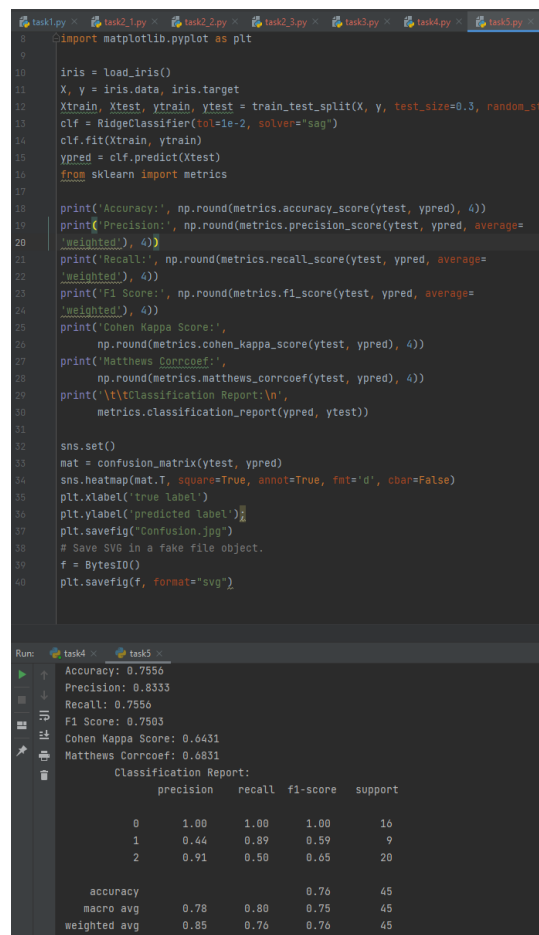


Рисунок 8. Класифікатор Ridge

		Горелко О. В.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.05.000 – Лр2	Арк.
		Пудеко І. В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

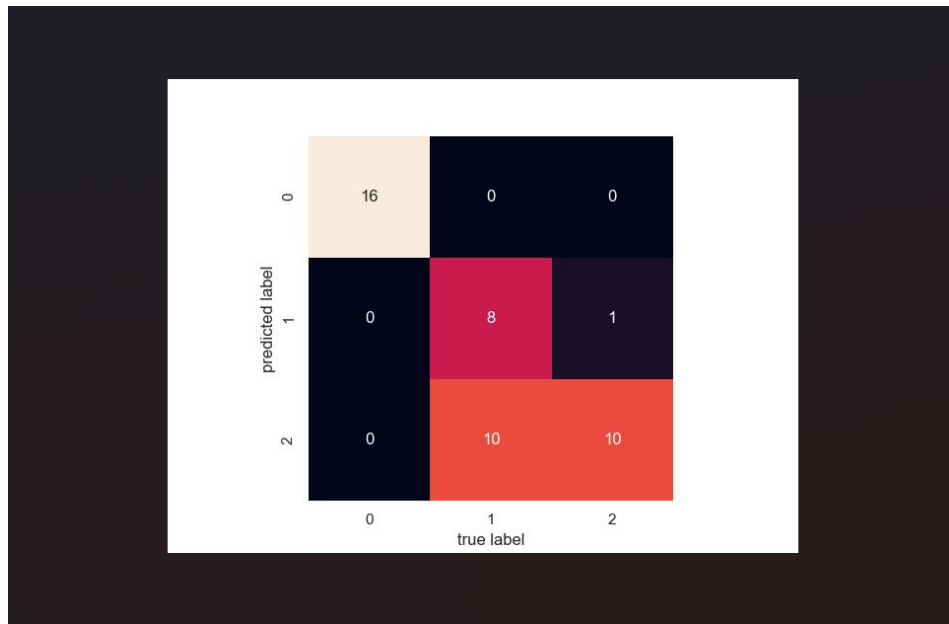


Рисунок 9. Confusion.jpg

Класифікатор має наступні параметри:

- tol – точність класифікації
- solver – алгоритм, який виконує класифікацію

На зображенні Confusion.jpg наведені результати класифікації. На вертикальній шкалі відкладені наявні класи ірису в числовій репрезентації, а на горизонтальній передбачення класу ірису. Цифра на перетині – кількість результатів системи при справжньому і передбаченому класі.

Коефіцієнт кореляції Метьюза – коефіцієнт, який на основі матриці помилок вираховує коефіцієнт від -1 до 1, де 1 – є результатом ідеальної класифікації, а 0 – рівень випадкового вибору.

Коефіцієнт Коена Каппа – коефіцієнт, який також за основу бере матрицю помилок, але замість загальної якості, звертає увагу на нерівноцінне розподілення класів.

Висновок: використовуючи спеціалізовані бібліотеки та мову програмування Python дослідив різні методи класифікації даних та навчився їх порівнювати.