МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 3

з дисципліни “Компоненти програмної інженерії”
тема “Ознайомлення з машинним навчанням”

Виконав студент
І курсу групи КП-01
Тітов Єгор Павлович

Перевірив
“____” “_____” 20___р
викладач
Радченко Костянтин
Олександрович

Варіант №17

Київ 2021

Мета

Ознайомитись з основними пакетами, які використовуються для машинного навчання в програмах, написаних мовою Python.

Навчитися розробляти сучасні інтелектуальні системи з використанням методів машинного навчання.

Завдання

Розробити програмне забезпечення для розпізнавання рукописних цифр.

Алгоритм розв'язання задачі

1. Візьмемо за приклад набір рукописних цифрових даних, використовуючи datasets з бібліотеки sklearn.
2. Створимо класифікатор, використовуючи svm з бібліотеки sklearn. Далі навчимо модель, використовуючи дані прикладу.
3. Введемо очікувані дані прогнозу, користуючись даними прикладу.
4. Знайдемо і виведемо прогнозовані значення

Текст програми

```
import matplotlib.pyplot as plt
from sklearn import datasets, svm, metrics

# Завантаження даних
digits = datasets.load_digits()
images_and_labels = list(zip(digits.images, digits.target))

# Виведення даних тренування
for index, (image, label) in enumerate(images_and_labels[:8]):
    plt.subplot(2, 8, index + 1)
    plt.axis('off')
    plt.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    plt.title('Train: %i' % label)

# Попередня обробка даних
n_samples = len(digits.images)
data = digits.images.reshape((n_samples, -1))

# Навчання і тренування моделі розпізнавання даних
classifier = svm.SVC(gamma=0.001)
classifier.fit(data[:n_samples // 2], digits.target[:n_samples // 2])
```

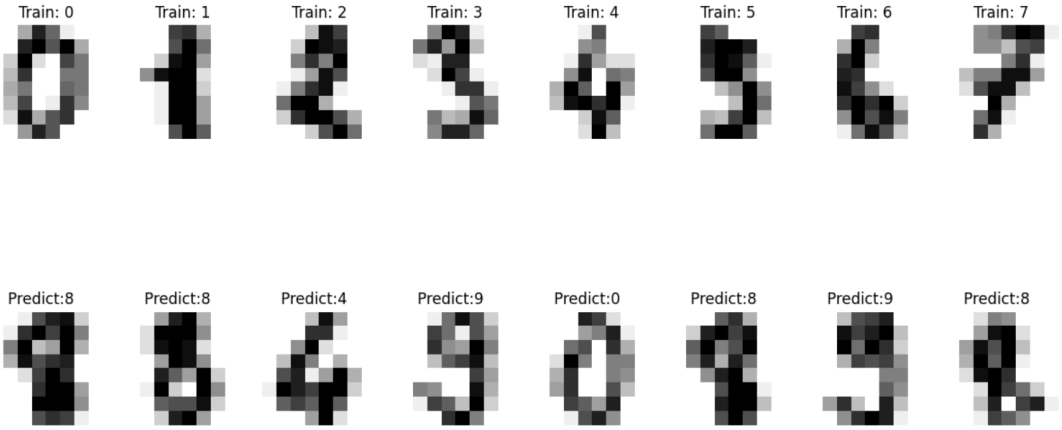
```
# Прогнозовані значення
expected = digits.target[n_samples // 2:]
predicted = classifier.predict(data[n_samples // 2:])

# Виведення класифікатора тесту і матриці неточностей
print("Classification report for classifier %s:\n%s\n" % (classifier,
metrics.classification_report(expected, predicted)))
print("Confusion matrix:\n%s" % metrics.confusion_matrix(expected, predicted))

# Виведення прогнозованих даних
images_and_predictions = list(zip(digits.images[n_samples // 2:], predicted))
for index, (image, prediction) in enumerate(images_and_predictions[:8]):
    plt.subplot(2, 8, index + 9)
    plt.axis('off')
    plt.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    plt.title('Predict:%i' % prediction + "  ")

plt.show()
```

Результати тестування



Classification report for classifier SVC(gamma=0.001):

	precision	recall	f1-score	support
0	1.00	0.99	0.99	88
1	0.99	0.97	0.98	91
2	0.99	0.99	0.99	86
3	0.98	0.87	0.92	91
4	0.99	0.96	0.97	92
5	0.95	0.97	0.96	91
6	0.99	0.99	0.99	91
7	0.96	0.99	0.97	89
8	0.94	1.00	0.97	88
9	0.93	0.98	0.95	92
accuracy			0.97	899
macro avg	0.97	0.97	0.97	899
weighted avg	0.97	0.97	0.97	899

Confusion matrix:

```
[[87  0  0  0  1  0  0  0  0  0]
 [ 0 88  1  0  0  0  0  0  1  1]
 [ 0  0 85  1  0  0  0  0  0  0]
 [ 0  0  0 79  0  3  0  4  5  0]
 [ 0  0  0  0 88  0  0  0  0  4]
 [ 0  0  0  0  0 88  1  0  0  2]
 [ 0  1  0  0  0  0 90  0  0  0]
 [ 0  0  0  0  0  1  0 88  0  0]
 [ 0  0  0  0  0  0  0  0 88  0]
 [ 0  0  0  1  0  1  0  0  0 90]]
```

Проблеми

Не виявлено.

Висновки

В процесі виконання лабораторної роботи було проведено ознайомлення з основними пакетами, які використовуються для машинного навчання в програмах, написаних мовою Python.

Також була пророблена робота з розроблення сучасних інтелектуальних систем з використанням методів машинного навчання.

Контрольні питання

1. ***Для чого призначена бібліотека pandas?***

Пакет pandas забезпечує аналіз даних з реального світу для програм, написаних мовою Python.

2. ***Які структури даних та для чого використовуються в пакеті pandas?***

Структурами даних, які використовуються в даному пакеті, є:

– Series ([data, index, dtype, name, copy, ...]) – одновимірний проіндексований масив, який може містити дані будь-яких типів (s = Series(data, index=index), де data може бути даними типу dict (Series працює подібно словникам), ndarray (Series працює подібно масивам NumPy) або скалярного типу, а index – перелік міток для даних);

– DataFrame ([data, index, columns, dtype, copy]) – двовимірний проіндексований масив з колонками потенційно будь-якого типу. DataFrame в якості даних може використовувати dict з одномірних ndarray, двомірні ndarray, структуровані ndarray або типу запис, Series, DataFrame;

– Panel ([data, items, major_axis, minor_axis, ...]) – контейнер для тривимірних даних;

– Panel4D ([data, labels, items, major_axis, ...]) – контейнер для чотирьохвимірних даних;

– PanelND – модуль з множиною фабричних функцій, який дозволяє користувачу створювати N-вимірні контейнери даних.

3. ***Які функції бібліотеки pandas можуть бути використані для введення/виведення даних?***

Для роботи з пропущеними значеннями в даних, які можуть бути представлені як NaN, None, inf, -inf, призначені функції: isnull(), notnull(), для заповнення пропущених значень – fillna ([value, method, axis, ...]), для вилучення індексів з пропущеними значеннями – dropna ([axis, inplace]), для інтерполяції пропущених значень – interpolate ([method, axis, limit, ...]).

4. ***Яким чином можна модифікувати дані, що зберігаються в структурах даних пакету pandas?***

Для роботи з даними структурами можуть використовуватися методи head ([n]) і tail ([n]), index (рядки), columns (для DataFrame), values для доступу до даних. DataFrame має методи add, sub, mul, div, пов'язані функції radd, rsub для виконання бінарних операцій і методи бінарного порівняння eq, ne, lt, gt, le, ge.

Для того щоб застосувати визначену користувачем функцію до двох структур даних типу DataFrame, необхідно використати метод combine, для виконання дій над однією структурою призначений метод apply (func[, axis]), у випадку необхідності роботи з невекторизованими даними використовуються методи applymap (func) і map (arg[, na_action]). Для підбиття статистичних підсумків над даними може використовуватися метод describe ([percentile_width, ...]).

5. ***Яким чином виконується індексація даних у пакеті pandas?***

Таблиця 3.1 – Операції індексації у DataFrame

Операція	**Синтаксис**	**Результат**
Вибрати стовпчик	df[col]	Series
Вибрати рядок за міткою	df.loc[label]	Series
Вибрати рядок за цілочисельною позицією	df.iloc[loc]	Series
Виконати зріз рядків	df[5:10]	DataFrame
Вибрати рядки за вектором логічних значень	df[bool_vec]	DataFrame

6. ***Які функції для роботи з декількома структурами даних підтримуються бібліотекою pandas?***

Функція concat(objs[, axis, join, join_axes, ...]) призначена для конкатенації об'єктів вздовж однієї з осей, де objs – перелік або словник об'єктів Series, DataFrame, Panel.

Для об'єднання двох DataFrame об'єктів призначена функція merge(left, right[, how, on, left_on, ...]), яка за суттю близька до аналогічної функції в SQL. Якщо ключі співпадають, то результат обчислюється у вигляді декартового добутку.

Для об'єднання двох об'єктів DataFrames з різними індексами використовується метод join (other[, on, how, lsuffix, ...]).

7. ***Для виконання яких завдань призначена бібліотека scikit-learn?***

Бібліотека scikit-learn – бібліотека машинного навчання, яка підтримує алгоритми класифікації, регресії, кластеризації.

8. ***Які засоби попереднього оброблення даних надає бібліотека scikit-learn?***

Модуль sklearn.preprocessing включає методи масштабування, центрування, нормалізації та перетворення у двійкову форму, зокрема:
– Binarizer([threshold, copy]) – перетворює у двійкову форму за порогом;
– Normalizer([norm, copy]) – нормалізує вибірки індивідуально до одиничної норми.

9. ***Які засоби кластеризації надає бібліотека scikit-learn?***

Модуль sklearn.cluster дозволяє виконувати кластеризацію непроіндексованих даних. Функції модуля:
– estimate_bandwidth(X[, quantile, ...]) – оцінює ширину полоси для використання з алгоритмом здвигу середнього;
– k_means(X, n_clusters[, init, ...]) – алгоритм кластеризації K- середніх.

10. ***Які засоби класифікації надає бібліотека scikit-learn?***

Модуль sklearn.naive_bayes імплементує алгоритм наївної байєсівської класифікації, включаючи наступні класи: GaussianNB, для поліноміальних моделей MultinomialNB([alpha, ...]), для багатомірних моделей Бернуллі BernoulliNB([alpha, binarize, ...]).

Модуль sklearn.multiclass імплементує наступні алгоритми багатокласової класифікації за допомогою відповідних класів та функцій підбору стратегії і прогнозування:
– OneVsRestClassifier(estimator[, ...]), fit_ovr(estimator, X, y[, n_jobs]), predict_ovr(estimators, ...) – багатокласова стратегія «один- проти-інших»;

– OneVsOneClassifier(estimator[, ...]), fit_ovo(estimator, X, y[, n_jobs]), predict_ovo(estimators, classes, X) – багатокласова стратегія «один-проти-одного».