|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Лабораторна робота №1**  **ПОПЕРЕДНЯ ОБРОБКА ТА КОНТРОЛЬОВАНА КЛА- СИФІКАЦІЯДАНИХ**  ***Мета роботи:*** використовуючи спеціалізовані бібліотеки та мову програ- мування Python дослідити попередню обробку та класифікацію даних.  **Завдання №2.1.1 – 2.1.4:**  **Код програми:**  import numpy as np from sklearn import preprocessing  input\_data = np.array([[5.1, -2.9, 3.3],  [-1.2, 7.8, -6.1],  [3.9, 0.4, 2.1],  [7.3, -9.9, -4.5]]) # Бінаризація даних data\_binarized = preprocessing.Binarizer(threshold=2.0).transform(input\_data) print(f"\nBinarized data:\n{data\_binarized}")  # Виведення середнього значення та стандартного відхилення print("\nBEFORE: ") print(f"Mean = {input\_data.mean(axis=0)}") print(f"Std deviation = {input\_data.std(axis=0)}")  # Исключение среднего data\_scaled = preprocessing.scale(input\_data) print("\nAFTER: ") print(f"Mean = {data\_scaled.mean(axis=0)}") print(f"Std deviation = {data\_scaled.std(axis=0)}")  # Масштабування MinМax data\_scaled\_minmax = preprocessing.MinMaxScaler(feature\_range=(0,1)) data\_scaled\_minmax = data\_scaled\_minmax.fit\_transform(input\_data) print(f"\nMin max scaled data:\n{data\_scaled\_minmax}")  # Нормалізація даних data\_normalized\_l1 = preprocessing.normalize(input\_data, norm='l1') data\_normalized\_l2 = preprocessing.normalize(input\_data, norm='l2')  print(f"\nL1 normalized data:\n{data\_normalized\_l1}") print(f"\nL2 normalized data:\n{data\_normalized\_l2}") | | | | | | | | | | |
| R |  |  |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | | | | | |
|  |  |  |  |  |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |
| *Розроб.* | | *Решетнюк Д. Р.* |  |  | Звіт з лабораторної роботи | *Літ.* | | | *Арк.* | *Аркушів* |
| *Перевір.* | | *Голенко М.Ю.* |  |  |  |  |  | *1* | *18* |
| *Керівник* | |  |  |  | *ФІКТ Гр. IПЗ-20-2[1]* | | | | |
| *Н. контр.* | |  |  |  |
| *Зав. каф.* | |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 2. Завдання 2.1 – 2.14  L1 – нормалізація використовує метод найменших абсолютних відхилень,  забезпечуючи рівність суми абсолютних значень в кожному рядку, яка дорівнює 1.  L2- нормалізація методом найменших квадратів забезпечує, що сума квадратів значень дорівнює 1, підсилюючи вплив великих значень ознак. Вона меньше чутлива до викидів порівняно з L1-нормалізацією. | | | | | | |
|  |  | *Решетнюк Д.Р..* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *2* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Завдання 2.1.5:**  **Код програми:**  import numpy as np from sklearn import preprocessing  input\_labels = ['red', 'black', 'red', 'green', 'black', 'yellow', 'white']  # Створення кодувальника та встановлення відповідності між мітками та числами encoder = preprocessing.LabelEncoder() encoder.fit(input\_labels)  # Виведення відображення print("\nLabel mapping: ") for i, item in enumerate(encoder.classes\_):  print(item, '-->', i)  # перетворення міток за допомогою кодувальника test\_labels = ['green', 'red', 'black'] encoded\_values = encoder.transform(test\_labels) print(f"\nLabels = {test\_labels}") print(f"Encoded values = {list(encoded\_values)}")  # Декодування набору чисел за допомогою декодера encoded\_values = [3, 0, 4, 1] decoded\_list = encoder.inverse\_transform(encoded\_values) print(f"\nEncoded values = {encoded\_values}") print(f"Decoded labels = {list(decoded\_list)}")    Рис. 4. Результат виконання | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *3* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Спочатку кодувальник було навчено на основі вхідних даних, використовуючи метод “.fit()”. Під час навчання кодувальник встановлює відповідність між мітками та числами, 'black 🡪 0, ‘grn 🡪 1’ і так далі.  Після цього створюється тестовий набір міток, і за допомогою методу “.transform()” тестові мітки кодуються у відповідні числа ‘green = 1’, ‘red = 2’, ‘black = 0’.  Щоб декодувати значення назад у початкові мітки, використовується метод “.inverse\_transform()”. Наприклад, ‘3 = white’, ‘0=black’ і так далі.  **Завдання 2.2:**    Рис. 5. Дані для варіанту    Рис. 6. Код програми | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *4* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 7. Результат виконання | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *5* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Завдання 2.3:**    Рис. 8. Код програми    Рис. 9. Результат виконання | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *6* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Завдання 2.4:**    Рис. 10. Код програми | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *7* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 11. Результат виконання    Рис. 12. Результат виконання | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *8* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Покращення методу обчислення шляхом перехресної перевірки.    Рис. 13. Код програми    Рис. 14. Результат виконання | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *9* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 15. Результат виконання  Обидва кодових відрізки використовують класифікатор наївного байєса для класифікації даних, проте перший навчає та тестує модель на одному наборі даних, тоді як другий робить розбивку на тренувальний і тестовий набори, проводячи при цьому перехресну перевірку для оцінки точності моделі на різних наборах даних.  Точність в першому кодовому відрізку обчислюється на тих же даних, на яких модель навчалася. Це може викликати завищення точності і не відображати реальну здатність моделі до узагальнення на нові дані.  У другому кодовому відрізку точність обчислюється на тестовому наборі, який не використовувався для навчання моделі. Це надає більш об'єктивну оцінку здатності моделі до прогнозування на нових даних.  Отже, точність в другому кодовому відрізку є більш інформативною і показує, наскільки добре модель навчається та узагальнюється на нові дані. | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *10* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Завдання 2.5:**    Рис. 16. Код програми | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *11* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 17. Код програми | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *12* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 18. Код програми | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *13* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 19. Результат виконання | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *14* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 20. Результат виконання roc\_curve | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *15* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 21. Результат виконання з AUC(аналіз продуктивності) доданою до легенди Висновок який можна дати отримавши ці дані:  Порівнюючи наївний байєсівський класифікатор та метод опорних векторів (SVM), було виявлено різниці в їхніх параметрах, зокрема в виборі функцій ядра. Обидва алгоритми є чутливими до оптимізації параметрів, і зміна цих параметрів може суттєво впливати на їх вихід. Результати, що показують перевагу одного алгоритму над іншим, можуть бути конкретними для обраних параметрів, і зміна параметрів може призвести до зміни ефективності кожного з них. | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *16* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Завдання 2.6:**    Рис. 22. Код програми | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *17* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рис. 23. Результат виконання  Accuracy, Precision, Recall та F1 для обох моделей дуже близькі, що свідчить про те, що обидві моделі розпізнають дані важливої якості майже однаковому рівні. Модель SVM має трохи вищу точність та кращі значення повноти для класів.  Важливо також звертати увагу на інші аспекти, такі як обсяг даних, обраний набір ознак та розмір навчального та тестового наборів, оскільки ці фактори можуть впливати на ефективність моделей.  https://github.com/cpacemon/ai\_Lab\_reshetnyuk | | | | | | |
|  |  | *Решетнюк Д.Р.* |  |  | *ДУ «Житомирська політехніка».23.121.4.000 – Лр1* | *Арк.* |
|  |  | *Голенко М.Ю.* |  |  | *18* |
| *Змн.* | *Арк.* | *№ докум.* | *Підпис* | *Дата* |