УДК 004.4
DOI https://doi.org/10.32782/tnv-tech.2023.2.4

ПЕРВИННА ОБРОБКА ДАНИХ РЕЗУЛЬТАТІВ СПОРТИВНИХ ТРЕНУВАНЬ СТУДЕНТІВ НА ВЕСЛУВАЛЬНИХ ТРЕНАЖЕРАХ CONCEPT2 ДЛЯ ПОДАЛЬШОГО АНАЛІЗУ ЗА ДОПОМОГОЮ БІБЛІОТЕКИ PANDAS

Горбань Г. В. *– кандидат технічних наук,*
доцент кафедри інженерії програмного забезпечення
Чорноморського національного університету імені Петра Могили
ORCID ID: 0000-0002-6512-3576
Scopus Author ID: 57103674400

Кандиба І. О. *– Ph.D.,*
старший викладач кафедри інженерії програмного забезпечення
Чорноморського національного університету імені Петра Могили
ORCID ID: 0000-0002-8589-4028
Scopus Author ID: 57212577217

Фісун М. Т. *– доктор технічних наук,*
професор кафедри інженерії програмного забезпечення
Чорноморського національного університету імені Петра Могили
ORCID ID: 0000-0003-1297-6230
Scopus Author ID: 57103586600

У статті представлено початковий етап дослідження спортивних результатів з академічного веслування студентів Чорноморського національного університету імені Петра Могили, який полягає у отриманні даних результатів тренування на веслувальних тренажерах Concept2 та подальшому співставленні з даними протоколів виконання веслувальних тестів. Наведено структуру даних, що зберігається у внутрішній пам'яті моніторів веслувальних тренажерів та вказано їх незручність при подальшій обробці та аналізі. Наведено використання при очищенні даних бібліотеки Pandas мови Python та її класу DataFrame, що представляє зручний спосіб зберігання даних у табличному вигляді та їх перетворенні. Детально наведено процес отримання кінцевого датафрейму, придатного для подальшого аналізу даних та застосуванні на них методів машинного навчання для виявлення залежностей між антопологічними характеристиками студентів та пройдених ними відстаней та потужностей гребків впродовж тренування, а також для прогнозування майбутніх спортивних результатів. Показано, що у початкових даних тренувань на веслувальних тренажерах зберігаються тренування протягом одного календарного року, а також що результати одного тренування представляються в декількох рядках, тому необхідно спочатку відфільтрувати тільки необхідні дані, а потім їх представити у вигляді, який у одному конкретному рядку зберігає дані тільки одного тренування. Наведено варіант реалізації цього представлення засобами бібліотеки Pandas. Представлено структуру даних протоколів виконання веслувальних тестів, що внесені викладачами з фізичного виховання, а саме наведено антропологічні характеристики студентів, що вносились у відповідні протоколи. Наведено процес отримання датафрейму з даними протоколів, який є результатом об'єднання двох аркушів файлу з протоколами. У якості останньої операції при очищенні даних представлено об'єднання двох отриманих датафреймів із даними веслувальних тренажерів та протоколів відповідно за загальними полями, якими є дата та час початку тренування та інвентарний номер тренажеру, що є аналогом операції JOIN у мові SQL.

Ключові слова: *спорт, академічне веслування, тренажер, Concept2, набір даних, Pandas, DataFrame, характеристика, об'єднання.*

Horban H. V., Kandyba I. O., Fisun M. T. Primary data processing of students' sports training results on rowing simulators Concept2 for further analysis using the Pandas library

The article presents the initial stage of the study of sports results in academic rowing of students of Petro Mohyla Black Sea National University, which consists in obtaining data on the results of training on rowing simulators Concept2 and further comparison with the data of rowing test protocols. The structure of the data stored in the internal memory of rowing simulator monitors is presented and their inconvenience in further processing and analysis is indicated. The use of the Pandas library of the Python language and its DataFrame class, which is a convenient way to store data in tabular form and transform it, is shown when cleaning data. The process of obtaining the final data frame suitable for further data analysis and applying machine learning methods to it to identify dependencies between the anthropological characteristics of students and the distances they covered and the power of their strokes during training, as well as to predict future sports results, is presented in detail. It is shown that the initial data of training on rowing simulators stores training during one calendar year, and that the results of one training session are presented in several lines, so it is necessary to first filter only the necessary data and then present them in a form that stores only one training session in one specific line. The article presents a variant of realization of this representation by means of the Pandas library. The data structure of the rowing test protocols entered by physical education teachers is presented, namely, the anthropological characteristics of students entered in the relevant protocols. The process of obtaining a dataframe with protocol data, which is the result of merging two sheets of a file with protocols, is presented. As the last operation in data cleaning, we present the union of the two obtained dataframes with the data of rowing machines and protocols, respectively, by common fields, which are the date and time of the start of training and the inventory number of the machine, which is analogous to the JOIN operation in SQL.

Key words: *sport, academic rowing, simulator, Concept2, dataset, Pandas, DataFrame, characteristics, association.*

Вступ. Розвиток сучасного спорту невід'ємно пов'язаний з науковими дослідженнями та впровадженням технологічних новинок у тренувальний процес спортсменів. спортсменів. Після кількох років кризових явищ в усіх суспільних сторонах життя, зокрема й у спорті, спостерігається підйом. Кожен новий етап у розвитку будь-якого виду спорту вимагає якісно нового вирішення цих завдань. Сукупність технічної, фізичної та інших сторін підготовленості спортсменів циклічних видів спорту, підпорядкована одній меті – досягненню можливої більшої швидкості на змаганні. Стрімке зростання спортивних результатів серед спортсменів різних країн передбачає серйозне вдосконалення тренувальних програм з урахуванням останніх досягнень спортивної науки.

У свою чергу, моделювання є достатньо важливим фактором організації та планування підготовки спортсмена, що дає можливість прогнозування бажаного рівня досягнення, правильного формулювання завдання та використання найбільш ефективних засобів тренування.

Створення та впровадження систем, що будуються на новітніх інформаційних технологіях і складних методах обробки даних, стають все більш важливими для збору, передачі, зберігання та аналізу даних з різних датчиків у спорті.

Аналіз останніх досліджень і публікацій. До сучасних методів штучного інтелекту, які використовують у спорті, належать кластерний аналіз, алгоритми регресії, метод опорних векторів, метод K-найближчих сусідів, асоціативні правила, нейронні мережі, методи нечіткої логіки, які використовуються для класифікації, кластеризації і прогнозування конкретних спортивних даних. Нині, зокрема, аналіз даних за допомогою методів самонавчання дедалі частіше обговорюються як перспективний напрям застосування у спортивній науці [1–3]. Комп'ютерні системи з концепцією нечіткої логіки, що застосовуються в спорті, включають у себе зібрані дані від пристроїв із датчиками, а також рекомендовані пропозиції з критеріями належного виконання вправ. Кінцева мета полягає в об'єднанні

розроблених процедур в комп'ютеризовану тренувальну систему з автоматизованим зворотним зв'язком за виконуваною методикою [4].

Фахівцями політехнічного університету Картахени та університету Віго (Іспанія) запропоновано динамічний програмний підхід для інтелектуальних платформ у бігових дисциплінах на основі марківських процесів прийняття рішень [5]. Він дозволяє спортсменам виконувати різнорідні тренувальні програми з кількома рівнями інтенсивності вправ.

Вчені університету Оулу (Фінляндія) реалізували можливість автоматичного розпізнавання рухової активності на смартфонах на основі даних акселерометра [6]. На відміну від більшості інших досліджень, було зібрано не тільки дані, використовуючи акселерометр смартфона, а й було реалізовано моделі розпізнавання активності в телефоні, також програмно було реалізовано весь процес класифікації (первинне опрацювання, виділення ознак і систематизація).

У цілому за висновками експертів поєднання кількох технологій є найбільш ефективним для отримання доступу до всіх значущих параметрів, оптимізуючи результативність спортсмена. У додаток до цифрових і статистичних методів, методи нейронних мереж, інтелектуального аналізу даних, нечіткої логіки, розпізнавання образів виявилися перспективними методами оцінювання та отримання інформації в спорті.

Постановка задачі. Академічне веслування – вид спорту з переважним проявом витривалості, у якому антропометричні дані і маса тіла є перевагою. Результативність змагальної діяльності обумовлюється елементами, що є специфічними для академічного веслування. У циклічних видах спорту, що пов' язані з проявом витривалості, найбільше значення мають рівномірність проходження різних відрізків дистанції та рівень дистанційної швидкості.

Метою дослідження є підвищення ефективності системи фізичного виховання студентів з академічного веслування як виду спорту в Чорноморському національному університеті імені Петра Могили шляхом побудови та використання у навчальному процесі системи обробки даних фізичної підготовленості, розвитку та функціональних можливостей студентів з подальшим аналізом даних, який буде мати можливість виявлення прихованих закономірностей та взаємозв'язків, а також прогнозування майбутніх спортивних результатів.

Виклад основного матеріалу дослідження. На сьогоднішній день при підготовці спортсменів високого класу в академічному веслуванні в більшості країнах світу використовують веслувальний ергометр Concept2 [7]. Наразі ергометри застосовуються при тестуванні різних сторін спеціальної фізичної та функціональної підготовленості веслярів. Спортсменам та тренерам надається можливість аналізу компонентів тренувального навантаження, відстеження кількості гребків, потужності виконання рухової дії, часу та інших параметрів. За допомогою моделювання веслувальні ергометри набувають все більш поширеного значення.

Щодо вивчення можливостей використання 6-хвилинного веслування на ергометрі Concept2 для оцінки рівня витривалості у студентів визначалось, що планувалась розробка бази даних з використанням комп'ютеризованих веслувальних ергометрів. Її розробка дозволила не лише зберігати, але й якісно обробляти інформацію, спрощуючи аналіз динаміки розвитку витривалості, силових якостей студентів. Для проєктування бази даних було використано протоколи виконання тестових вправ на веслувальному ергометрі, що були заповнені вручну викладачами кафедри теорії та методики фізичного виховання (рис. 1). Також було використано дані тренувань, що зняті безпосередньо з веслувальних тренажерів.

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID	Date	Time	ConceptNumber	Name	Sex	Distance	Age	HeartRate	HeartRate	Height	Weight
2	1	29,11	16,01	5	[redacted]	ч	1547	20	120	180	193	75
3	2	29,11	9,45	5	[redacted]	ч	1431	19	130	190	181	120
4	3	29,11	10,59	5	[redacted]	ч	1346	19	100	190	181	72
5	4	29,11	12,21	8	[redacted]	ч	1690	20	80	180	196	76
6	5	29,11	12,21	5	[redacted]	ч	1463	19	130	200	173	73
7	6	29,11	13,00	5	[redacted]	ж	1109	19	110	210	165	51
8	7	29,11	13,00	8	[redacted]	ж	1127	19	110	190	171	51
9	8	2,12	14,44	6	[redacted]	ж	1011	20	110	160	170	54
10	9	2,12	14,44	8	[redacted]	ж	1093	21	100	170	171	65
11	10	2,12	15,02	8	[redacted]	ч	1349	17	120	180	176	58
12	11	2,12	15,02	5	[redacted]	ч	1492	20	80	190	182	76
13	12	2,12	15,23	5	[redacted]	ж	1162	18	100	160	166	50
14	13	2,12	15,23	8	[redacted]	ж	1220	18	100	200	163	61
15	14	2,12	15,33	8	[redacted]	ч	1325	21	110	180	184	70
16	15	2,12	16,02	6	[redacted]	ж	1116	17	90	170	159	52
17	16	2,12	16,02	8	[redacted]	ч	1155	19	100	180	167	52
18	17	2,12	16,12	8	[redacted]	ч	1544	20	100	200	183	78
19	18	2,12	16,12	6	[redacted]	ч	1563	20	120	190	183	79
20	19	2,12	12,28	8	[redacted]	ч	1336	17	120	220	177	57
21	20	2,12	12,28	6	[redacted]	ж	1406	17	100	200	173	66
22	21	2,12	12,21	6	[redacted]	ч	1432	19	120	190	185	65
23	22	2,12	12,21	8	[redacted]	ч	1499	17	130	200	175	65
24	23	2,12	12,21	5	[redacted]	ч	1476	18	120	180	178	71
25	24	2,12	12,54	6	[redacted]	ж	1233	20	100	170	172	52
26	25	2,12	12,54	8	[redacted]	ж	1121	19	100	160	159	46
27	26	2,12	13,07	6	[redacted]	ч	1350	17	110	150	182	65
28	27	2,12	13,07	8	[redacted]	ч	1303	18	90	170	180	70
29	28	2,12	13,17	6	[redacted]	ч	1485	20	120	160	182	65

Рис. 1. Дані файлу протоколу проходження тестів на веслувальному тренажері

У протоколі визначаються такі дані як дата та час проходження тесту на веслувальному тренажері, номер тренажеру (у виконанні тесту було задіяно тренажери з номерами 5, 6 та 8), ім'я та прізвище студента (на рисунку ці дані закреслені через етичні міркування), стать, дистанція (повинна співпадати з результатом, отриманим на моніторі веслувального тренажеру), результати вимірювання пульсу до та після тренування, вага та зріст студента.

У свою чергу для фіксації та зберігання результатів тренувань на веслувальному тренажені використовується монітор PM5, що дає можливість накопичення та збереження даних за певний період часу. Існує функціонал обміну даними між монітором PM5 та персональним комп'ютером, відповідно до якого дані зберігаються в форматі csv. З монітору імпортуються такі дані як режим тренування, загальна відстань у метрах, пройдена за весь час тренування; середня кількість помахів весла при виконанні тренування; час виконання тренування, потужності помахів весла в калоріях на годину та ваттах за загальною відстаню. Приклад csv-файлу, у який імпортовано вказані вище дані та який відкритий у програмі Excel, наведено на рис. 2.

Як можна побачити, у файлі представлено дані певного тренування, що розділені на рівні часові відрізки. Це дає змогу отримати дані середньої кількості помахів, швидкості, часу подолання відстані 500 м, а також потужностей у калоріях на годину та ваттах за конкретний період часу тренування. У файл також імпортується дата та час початку тренування. Тому зіставивши дані дат і часу початку тренувань та пройдених відстаней, зазначених у протоколах, з імпортованими даними, можна отримати більш детальні дані виконання тесту кожного студента на веслувальному тренажері в різні періоди часу.

Однак основною проблемою є те, що монітор PM5 зберігає дані тренувань протягом всього року. Тому для того, щоб знайти детальні дані тренування по кожному студенту вручну, необхідно продивитись весь файл та зіставити відповідні дані з протоколами. Це може зайняти багато часу, оскільки при цьому необхідно

відкидати дані тренувань, що відбувались не у дні, що зазначені у протоколах. Також ще більшою проблемою є те, що у ті дні може бути що далеко не всі тренування на тренажері виконували студенти. Щоб зрозуміти, що дані певного тренування дійсно належали студенту, необхідно дивитись на колонку режиму. Для студентського тесту значення буде дорівнювати «0:06:00» що означає, що тест на тренажері проходив рівно 6 хвилини і дані представлено за 3 рівних відрізки часу: за 2, 4 та 6 хвилин.

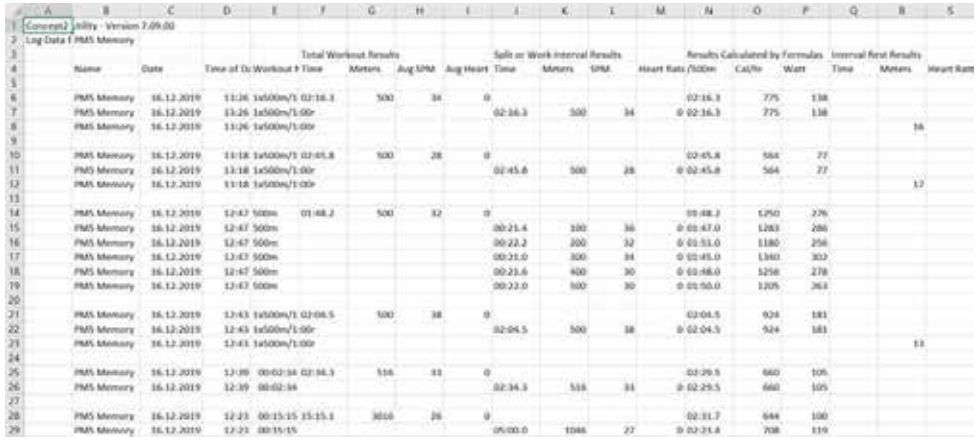

Рис. 2. Відкритий у програмі Excel csv-файл з імпортованими даними з монітору PM5

Для формування бази даних файли з імпортованими результатами тренувань у кількості трьох було переглянуто вручну, в тім на це було витрачено багато часу. Тому постало питання як можна автоматизувати цей процес, одразу отримавши тільки необхідні дані. Для вирішення цієї проблеми було досліджено бібліотеку Pandas мови програмування Python.

Pandas призначено для маніпулювання числовими таблицями та часовими рядами, а основною областю застосування є забезпечення збору та очищення даних, а також їх аналізу та моделювання. Головними структурами у Pandas є Series та DataFrame. Series представляє собою об’єкт, що схожий на одновимірний масив, однак його особливістю є наявність індексів уздовж кожного елементу зі списку. У свою чергу, DataFrame представляється у вигляді звичайної таблиці, у якій завжди присутні рядки та стовпці. У DataFrame стовпцями є об’єкти Series, рядки яких є їх безпосередніми елементами. До того ж Pandas підтримує всі найпопулярніші формати зберігання даних: csv, excel, sql, html тощо. Це робить можливість застосувати для очищення даних тренувань саме Pandas.

Для початку необхідно зчитати дані, що були імпортовані з моніторів трьох веслувальних тренажерів, що позначені номерами 5, 6 та 8, у відповідні датафрейми Pandas:

```
concept5Df = pd.read_csv("./LogBook5.csv”, sep=’;’, header = [3])
concept6Df = pd.read_csv("./LogBook6.csv”, sep=’;’, header = [3])
concept8Df = pd.read_csv("./LogBook8.csv”, sep=’;’, header = [3])
```

Об’єднавши всі датафрейми в один, буде отриманий єдиний датафрейм з усіма тренуваннями на тренажерах:

```
conceptDf = pd.concat([concept5Df, concept6Df, concept8Df])
```

На наступному етапі необхідно у отриманому датафреймі залишити тільки результати студентських тестів на веслувальних тренажерах. Це можна зробити, залишивши у датафреймі тільки рядки, значення стовпця режиму тренування для яких дорівнює «0:06:00». Це ще не остаточно будуть тільки дані саме студентських тестів, зазначених у протоколі, оскільки можуть бути інші дані тренувань тривалістю 6 хвилин. Для цього спочатку приведемо тип даних у комірці режиму тренування до рядка (string), після чого відфільтруємо дані:

```
conceptDf['Workout Name'] = conceptDf['Workout Name'].
astype("string")
conceptDf = conceptDf[conceptDf['Workout Name'] == "0:06:00"]
```

Отриманий датафрейм представлено на рис. 3.

	Date	Time of Day	Time	Meters	Avg SPM	Time.1	Meters.1	SPM	Heart Rate	/500m	Cal/hr	Watt	Number
69	13.12.2019	04:55	06:00.0	1320.0	27.0	NaN	NaN	NaN	NaN	02:16.3	775.0	138.0	5
70	13.12.2019	04:55	NaN	NaN	NaN	02:00.0	429.0	28.0	0.0	02:19.8	740.0	128.0	5
71	13.12.2019	04:55	NaN	NaN	NaN	04:00.0	419.0	26.0	0.0	02:23.1	710.0	119.0	5
72	13.12.2019	04:55	NaN	NaN	NaN	06:00.0	473.0	29.0	0.0	02:06.8	890.0	171.0	5
84	13.12.2019	03:15	06:00.0	1462.0	28.0	NaN	NaN	NaN	NaN	02:03.1	945.0	188.0	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...
749	29.11.2019	11:43	NaN	NaN	NaN	04:00.0	431.0	24.0	0.0	02:19.2	746.0	130.0	8
750	29.11.2019	11:43	NaN	NaN	NaN	06:00.0	369.0	25.0	0.0	02:42.6	580.0	81.0	8
3040	16.04.2019	15:30	06:00.6	984.0	28.0	NaN	NaN	NaN	NaN	03:03.2	495.0	57.0	8
3041	16.04.2019	15:30	NaN	NaN	NaN	05:00.0	807.0	27.0	0.0	03:05.8	487.0	55.0	8
3042	16.04.2019	15:30	NaN	NaN	NaN	06:00.6	177.0	35.0	0.0	02:51.1	540.0	70.0	8

Рис. 3. Датафрейм із загальними даними тренувань тривалістю 6 хвилин на усіх веслувальних тренажерах

Достатньо великою проблемою при подальшому приведенні даних до таких, що можуть бути пристосовані для подальшого аналізу та застосування методів машинного навчання, є те, що у датафреймі результати одного окремого тренування представляються у чотирьох розташованих поряд рядках. Відповідно у першому з них визначаються дані результатів тренування за весь час його виконання. Відповідно подібні рядки у значенні стовпця з назвою “Time” мають значення «06:00.0» на відміну від наступних трьох рядків, для яких значення стовпця “Time” є порожнім, натомість не є порожнім значення стовпця “Time.1», що визначає певний проміжний період часу. Другий рядок з даними одного тренування представляє результати тренування впродовж перших двох хвилин тренування і у стовпці “Time.1» має значення «02:00.0», натомість значення стовпця “Time” є порожнім. Аналогічно третій на четвертий рядки представляють результати тренування впродовж двох наступних хвилин (третьої та четвертої) та двох останніх хвилин (п’ятої та шостої) відповідно. Значення стовпців “Time.1» для цих рядків становитимуть «04:00.0» та «06:00.0».

Описана структура є досить важкою для подальшої обробки, тому необхідно відфільтрувати окремі рядки у датафрейми, а потім об’єднати їх в один. Для цього спочатку виділяємо датафрейм із загальними даними за весь час тренування, привівши до того значення стовпця “Time” до типу “string”:

```
conceptDf['Time'] = conceptDf['Time'].astype("string")
conceptFulltimeDf = conceptDf[conceptDf['Time'] == "06:00.0"]
```

Далі для того, щоб у подальшому об'єднати отриманий датафрейм з іншими, з нього потрібно видалити стовпці, що точно мають порожні значення (відповідно вони матимуть непорожні значення для датафреймів з даними тренувань за певні проміжні періоди часу):

```
conceptFulltimeDf = conceptFulltimeDf.drop(columns = ["Time.1",
"Meters.1","SPM","Heart Rate"])
```

Так само виділяємо в окремі датафрейми дані тренування за певні зазначені вище періоди, оброблюючи дані стовпчика "Time.1» у вхідному датафреймі:

```
conceptDf['Time.1'] = conceptDf['Time.1'].astype("string")
concept2MinDf = conceptDf[conceptDf['Time.1'] == "02:00.0"]
concept2MinDf = concept2MinDf.drop(columns=["Time","Meters",
"Avg SPM","Number"])
concept4MinDf = conceptDf[conceptDf['Time.1'] == "04:00.0"]
concept4MinDf = concept4MinDf.drop(columns=["Time","Meters",
"Avg SPM","Number"])
concept6MinDf = conceptDf[conceptDf['Time.1'] == "06:00.0"]
concept6MinDf = concept6MinDf.drop(columns=["Time","Meters",
"Avg SPM","Number"])
```

Перед об'єднанням всіх датафреймів в один результуючий маємо ще одну проблему, що полягає у однаковій назві певних стовпців для датафреймів з результатами за проміжні періоди. Тому необхідно задати таким стовпцям унікальні імена, виділивши у ньому конкретний період часу:

```
concept2MinDf = concept2MinDf.drop(columns=["Heart
Rate","Time.1"])
concept2MinDf = concept2MinDf.rename(columns = {'Meters.1':
'Meters_2min','SPM':'SPM_2min','/500m':'/500m_2min','Cal/hr':'Cal/
hr_2min','Watt':'Watt_2min'})
```

Такі самі дії виконаємо і для двох інших датафреймів з даними проміжних періодів.

Останнім етапом перед об'єднанням датафреймів є перевстановлення індексу у кожному датафреймі, щоб подальше об'єднання здійснилося правильно, та видалення стовпців зі старими індексами.

```
conceptFulltimeDf = conceptFulltimeDf.reset_index()
conceptFulltimeDf = conceptFulltimeDf.drop(columns="index")
concept2MinDf = concept2MinDf.reset_index()
concept2MinDf = concept2MinDf.drop(columns=["index","Date",
"Time of Day"])
```

Слід зазначити, що у датафреймах з результатами за окремі проміжки часу також були видалені стовпці з назвами "Date" та "Time of day" для запобігання стовпців-дублікатів, оскільки ті самі дані присутні у відповідних стовпцях датафрейму з результатами за загальний час.

Тепер можна об'єднати всі датафрейми методом concat для отримання одного результуючого. Назвемо його tempDf та видалимо у ньому стовпець "Time", оскільки до всіх рядків у ньому буде єдине значення («06:00.0»), тому він у подальшому вже не буде потрібний.

Датафрейм tempDf містить очищені дані виконання тесту на веслувальному тренажері Concept2, в якому один рядок представляє дані конкретного тренування

та містить всі необхідні для подальшого аналізу характеристики як для загальних результатів, так і для результатів за три проміжні відрізки часу відповідного тренування. Даний датафрейм представлено на рис. 4.

	Date	Time of Day	Meters	Avg SPM	/500m	Cal/hr	Watt	Number	Meters_2min	SPM_2min	...	Meters_4min	SPM_4min	/500m_4min	Cal/hr_4min	Watt_4m
0	13.12.2019	04:55	1320.0	27.0	02:16.3	775.0	138.0	5	429.0	28.0	...	419.0	28.0	02:23.1	716.0	119
1	13.12.2019	03:15	1462.0	28.0	02:03.1	945.0	188.0	5	439.0	28.0	...	502.0	28.0	01:59.5	1005.0	205
2	11.12.2019	02:50	1265.0	26.0	02:22.2	718.0	121.0	5	447.0	27.0	...	423.0	25.0	02:21.8	722.0	123
3	11.12.2019	02:36	1111.0	28.0	02:42.0	583.0	92.0	5	406.0	31.0	...	350.0	26.0	02:51.4	539.0	89
4	11.12.2019	02:22	1259.0	32.0	02:22.9	712.0	120.0	5	462.0	34.0	...	402.0	31.0	02:29.2	662.0	109
...																
195	29.11.2019	12:30	1680.0	29.0	01:46.5	1296.0	290.0	8	618.0	33.0	...	544.0	28.0	01:50.2	1167.0	261
196	29.11.2019	12:19	907.0	24.0	03:14.1	464.0	48.0	8	314.0	24.0	...	292.0	23.0	03:25.4	438.0	40
197	29.11.2019	12:08	1677.0	27.0	02:47.1	558.0	75.0	8	356.0	26.0	...	359.0	26.0	02:47.1	558.0	75
198	29.11.2019	11:57	1667.0	27.0	02:45.5	565.0	77.0	8	383.0	24.0	...	365.0	25.0	02:44.3	571.0	79
199	29.11.2019	11:43	1164.0	24.0	02:32.0	642.0	100.0	8	384.0	25.0	...	431.0	24.0	02:19.2	746.0	130

200 rows × 23 columns

Рис. 4. Датафрейм з очищеними даними студентських тренувань на всіх тренажерах

В тім, отриманий датафрейм ще не можна назвати результуючим, оскільки у ньому поки не присутні антропологічні дані студентів, що зазначені у протоколах.

У свою чергу дані протоколів проходження тестів на веслувальних тренажерах збережені у файлі формату xlsx, який має два аркуші, у яких зберігаються дані окремо українських та індійських студентів відповідно.

Подібно методу зчитування даних з файлу формату csv бібліотека Pandas підтримує і метод зчитування даних формату xlsx. Тоді зчитаємо дані з відповідних аркушів, додавши при цьому до кожного датафрейму стовпець зі значенням національності студента (“Nationality”):

```
protocolsUkrDf = pd.read_excel("./Protocols.xlsx”, sheet_name = “Ukrainians”)
protocolsUkrDf = protocolsUkrDf .assign(Nationality = “Ukrainian”)
protocolsIndDf = pd.read_excel("./Protocols.xlsx”, sheet_name = “Indians”)
protocolsIndDf = protocolsIndDf.assign(Nationality = “Indian”)
```

Оскільки обидва датафрейми мають однакову структуру, то їх можна об’єднати в один результуючий датафрейм методом concat:

```
protocolsDf = pd.concat([protocolsUkrDf,protocolsIndDf])
```

Після цього в отриманому датафреймі перевстановимо індекс та видалимо стопці “ID” та “Index”, оскільки вони більше не будуть потрібні.

```
protocolsDf = protocolsDf.reset_index()
protocolsDf = protocolsDf.drop(columns=[“index”,“ID”)
```

Об’єднаний датафрейм з даними протоколів представлено на рис. 5.

Таким чином, було отримано два датафрейми. Перший представляє дані результатів проходження тесту на веслувальних тренажерах, а другий – дані протоколів. Для отримання вихідного датафрейму, готового для застосування на ньому методів машинного навчання, у якому були б представлені абсолютно всі характеристики, потрібно об’єднати зазначені вище датафрейми в один. В тім ці

датафрейми мають абсолютно різну структуру. Також можна припустити, що вони можуть мати різну кількість рядків, оскільки при заповненні протоколів вручну має місце людський фактор, коли результати певного тренування могли бути не внесені у відповідний файл .xlsx. Також у датафреймі результатів тесту на веслувальних тренажерах можуть бути присутні зайві рядки через те, що у деякий момент часу могло мати місце тренування режиму тривалості 6 хвилин, яке не стосується проходження тестів студентами на веслувальному тренажері, що вказано у відповідному файлі протоколів. На попередньому етапі очищення даних такі рядки виявити такі рядки, а потім і видалити їх, було неможливо.

	Date	Time	ConceptNumber	Name	Sex	Distance	Age	HeartRateBefore	HeartRateAfter	Height	Weight
0	2019-11-29	16:01:00	5	[redacted]	ч	1547	20	120	180	193	75.0
1	2019-11-29	09:44:00	5	[redacted]	ч	1431	19	130	190	181	120.0
2	2019-11-29	10:59:00	5	[redacted]	ч	1346	19	100	190	181	72.0
3	2019-11-29	12:30:00	8	[redacted]	ч	1690	20	80	180	196	78.0
4	2019-11-29	12:21:00	5	[redacted]	ч	1463	19	130	200	173	73.0
...	...	...	...	...	...	...	...	...	...	...	...
191	2019-12-11	12:00:00	8	[redacted]	ч	1234	19	120	230	171	46.0
192	2019-12-11	11:50:00	5	[redacted]	ч	1111	19	120	230	171	46.0
193	2019-12-11	12:42:00	6	[redacted]	ч	1309	21	108	150	171	53.0
194	2019-12-11	12:14:00	8	[redacted]	ч	1382	21	120	150	169	62.0
195	2019-12-11	12:04:00	5	[redacted]	ч	1265	19	100	190	180	66.0

196 rows × 12 columns

Рис. 5. Кінцевий датафрейм з даними протоколів

Тому, для об’єднання двох зазначених датафреймів вже неможливо застосувати метод concat, оскільки для його застосування датафрейми повинні мати однакову кількість рядків. Щоб об’єднати датафрейми результатів тренувань на тренажерах та даних протоколів, більше підходить метод join, який виконує об’єднання двох датафреймів подібно до операції JOIN у мові SQL для об’єднання двох реляційних таблиць [8]. Для цього у датафреймах повинно бути одне або декілька спільних полів.

Такими полями можуть бути дата, час та номер тренажера. В тім, з полями дати можуть виникнути проблеми, що пов’язані з неправильним форматом. Якщо розглянути датафрейм із даними протоколів, то значення відповідного стовпця зчитались як рядок, те саме можна сказати і про час. Серед типів даних, що можуть бути використані у Pandas, є тип datetime64, який одночасно визначає дату та час. Тому далі доцільно сформувати новий стовпчик, що буде об’єднаним значенням дати та часу, а потім перетворити його у тип datetime64.

```
protocolsDf = protocolsDf.assign(Date_Time = protocolsDf.Date + " " + protocolsDf.Time)
protocolsDf[“Date_Time”]=protocolsDf[“Date_Time”].astype(“datetime64[ns]")
```

Те саме проробимо і з відповідними стовпцями сформованого до цього датафрейму результатів тренувань на веслувальних тренажерах. До того ж у ньому значення стовпця дати представлено в іншому форматі, що відрізняються від формату дати у файлі з даними протоколів. Тому спочатку приводимо значення стовпця “Date” до дати, вказавши відповідний формат, а потім здійснимо ті самі операції, що і у випадку датафрейму з даними протоколів.

```
tempDf['Date']=pd.to_datetime(tempDf['Date'],format='%d.%m.%Y').
dt.date
tempDf["Date"] = tempDf["Date"].astype("string")
tempDf["Time of Day"] = tempDf["Time of Day"].astype("string")
tempDf = tempDf.assign(Date_Time = tempDf["Date"] + " " +
tempDf["Time of Day"])
tempDf["Date_Time"]=tempDf["Date_Time"].astype("datetime64[ns]")
```

Далі було помічено, що один з тренажерів, а саме тренажер з інвентарним номером 5 мав неправильні установки часу, через що час на ньому відставав від правильного часу на 9 годин та 14 хвилин. Цей факт примусив виконувати додаткові перетворення, для яких довелося використати клас timedelta зі стандартної бібліотеки Python під назвою datetime:

```
from datetime import timedelta
delta = timedelta(hours=9, minutes=14)
```

Для застосування зміщення часу тимчасово довелось привести значення стовпця "Date_Time" до типу datetime з однойменної бібліотеки мови Python.

Інша проблема полягала у тому, що не для всіх рядків датафрейму результатів тренувань значення часу потрібно змінювати, а тільки для тих рядків, що відповідають результатам на тренажері 3 номером 5.

Для цього виділяємо тимчасові датафрейми з результатами на тренажері з номером 5 та на інших тренажерах. Назвемо відповідно ці датафрейми Concept5Df та Concept68Df. Далі для першого з датафреймів створюємо новий стовпець зі зміщеним часом, видалимо стовпець з новим часом та здамо новому стовпцю назву видаленого. Наприкінці об'єднуємо тимчасові датафрейм, отримавши новий датафрейм зі всіма результатами тренувань на тренажерах.

```
concept5Df = tempDf[tempDf["Number"] == "5"]
concept68Df = tempDf[tempDf["Number"] != "5"]
concept5Df = concept5Df.assign(TempDateTime = concept5Df.Date_
Time + delta)
concept5Df = concept5Df.drop(columns="Date_Time")
concept5Df=concept5Df.rename(columns={"TempDateTime":"Date_
Time"})
tempDf = pd.concat([concept5Df,concept68Df])
```

Останньою операцією є об'єднання отриманих датафреймів результатів тренування та протоколів. З'єднувальними стовпцями у цьому випадку будуть дата початку тренування та інвертарний номер тренажера. Другий стовпець необхідно включити, оскільки деякі тренування могли початись одночасно на різних тренажерах, тому в якості первинного ключа повинна бути сукупність дати та номера тренажера.

Для об'єднання було застосовано метод merge бібліотеки Pandas, який працює подібно операції JOIN у мові SQL:

```
resultDf = pd.merge(tempDf, protocolsDf, how="right", on=["Date_
Time","Number"])
```

В результаті було отримано кінцевий датафрейм, представлений на рис. 6.

Отриманий датафрейм містить всі необхідні дані та може бути використаний для подальшого аналізу та застосування методів машинного навчання для прогнозування майбутніх результатів.

	Date_Time	Meters	Avg SPM	/500m	Cal/hr	Watt	Number	Meters_2min	SPM_2min	/500m_2min	...	Watt_6min	Name	Sex	Distance	Age	HeartRate
0	2019-12-06 12:39:00	501	24	03:19.7	451	44	5	280	24	03:30.0	...	54	[redacted]	x	501.0	19.0	
1	2019-12-02 14:33:00	971	28	03:05.3	489	55	5	319	30	03:08.0	...	60	[redacted]	x	971.0	18.0	
2	2019-12-04 15:28:00	988	31	03:02.1	499	58	5	314	31	03:11.0	...	71	[redacted]	x	988.0	20.0	
3	2019-12-11 11:22:00	1019	25	02:56.8	517	63	5	384	27	02:36.2	...	47	[redacted]	x	1023.0	20.0	
4	2019-12-04 15:16:00	1023	26	02:55.9	521	64	5	330	27	03:01.8	...	61	[redacted]	x	1030.0	18.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
191	2019-12-02 12:36:00	1436	31	02:05.3	911	178	8	496	32	02:00.9	...	175	[redacted]	ч	1500.0	17.0	
192	2019-12-10 15:57:00	1437	29	02:05.2	912	176	8	508	26	01:56.1	...	171	[redacted]	ч	1520.0	18.0	
193	2019-12-02 09:39:00	1497	29	02:00.2	992	201	8	475	29	02:06.0	...	234	[redacted]	ч	1523.0	19.0	
194	2019-12-02 12:50:00	1499	30	02:00.0	995	202	8	538	31	01:51.5	...	176	[redacted]	ч	1544.0	20.0	
195	2019-12-09 13:30:00	1500	30	02:00.0	997	203	8	505	31	01:58.8	...	199	[redacted]	ч	1600.0	20.0	

196 rows × 31 columns

Рис. 6. Кінцевий датафрейм

Висновки та перспективи подальших досліджень. Таким чином, було здійснено обробку імпортованих даних тренувань на веслувальних тренажерах Concept2, що складається з очищення непотрібних даних та об’єднання необхідних даних з даними, зазначеними у протоколах виконання веслувальних тестів студентами ЧНУ імені Петра Могили. Описаний підхід буде використаний у подальших проведеннях тестів на веслувальних тренажерах після завершення воєнного стану. Структури даних, отримані в результаті обробки, планується використовувати для подальшого аналізу з виявлення залежностей між антропометричними даними студентів та їх результатами, показаними при виконання веслувальних тестів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. P. Lamb, R. Bartlett, A. Robins. Self-organizing maps: An objective method for clustering complex human movement. *International Journal of Computer Science in Sport*, 9(1), 2010. P. 20–29.

2. H. Ghasemzadeh, R. Jafari. Coordination analysis of human movements with body sensor networks: A signal processing model to evaluate baseball swings. *IEEE Sensors Journal*, 11(3), 2011. P. 603–610.

3. A. Baca. Methods for recognition and classification of human motion patterns-a prerequisite for intelligent devices assisting in sports activities. *IFACProceedings Volumes*, 45(2), 2012. P. 55–61.

4. H. Novatchkov, A. Baca. Fuzzy logic in sports: a review and an illustrative case study in the field of strength training. *International Journal of Computer Applications*, 71(6), 2013. P. 8–14.

5. J. Wang, R. Chen, X. Sun, M. F. She, etc. Recognizing human daily activities from accelerometer signal. *Procedia Engineering*, 15, 2011. P. 1780–1786.

6. K. Taylor, U. A. Abdulla, R. J. Helmer, J. Lee, etc. Activity classification with smart phones for sports activities. *Procedia Engineering*, 13, 2011. P. 428–433.

7. Concept2 Rowing Machine – RowErg with PM5 – By Direct. URL: https://www.concept2.com/indoor-rowers/concept2-rowerg (дата звернення: 23.05.23).

8. B. Bateman, S. Basak, T. Joseph, W. So. The Pandas Workshop. Packt Publishing, 2022. 744 p.

REFERENCES:

1. P. Lamb, R. Bartlett, A. Robins. Self-organizing maps: An objective method for clustering complex human movement. *International Journal of Computer Science in Sport*, 9(1), 2010. pp. 20–29.

2. H. Ghasemzadeh, R. Jafari. Coordination analysis of human movements with body sensor networks: A signal processing model to evaluate baseball swings. *IEEE Sensors Journal*, 11(3), 2011. pp. 603–610.

3. A. Baca. Methods for recognition and classification of human motion patterns-a prerequisite for intelligent devices assisting in sports activities. *IFACProceedings Volumes*, 45(2), 2012. pp. 55–61.

4. H. Novatchkov, A. Baca. Fuzzy logic in sports: a review and an illustrative case study in the field of strength training. *International Journal of Computer Applications*, 71(6), 2013. pp. 8–14.

5. J. Wang, R. Chen, X. Sun, M. F. She, etc. Recognizing human daily activities from accelerometer signal. *Procedia Engineering*, 15, 2011. pp. 1780–1786.

6. K. Taylor, U. A. Abdulla, R. J. Helmer, J. Lee, etc. Activity classification with smart phones for sports activities. *Procedia Engineering*, 13, 2011. pp. 428–433.

7. Concept2 Rowing Machine – RowErg with PM5 – By Direct. URL: https://www.concept2.com/indoor-rowers/concept2-rowerg (дата звернення: 23.05.23).

8. B. Bateman, S. Basak, T. Joseph, W. So. The Pandas Workshop. Packt Publishing, 2022. 744 p.