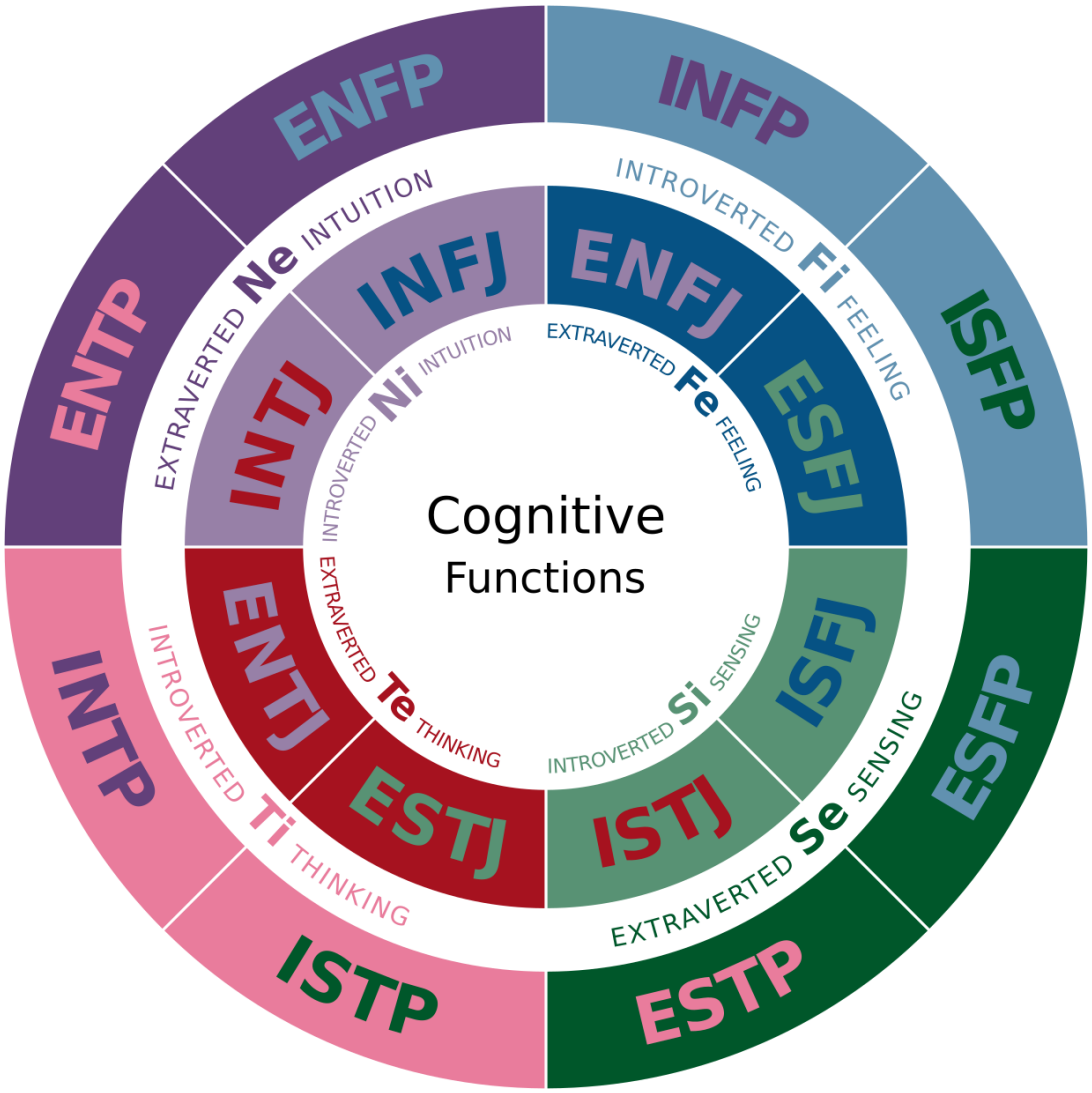
Projekt

Agata Margas, Daniel Barczyk

16 czerwca 2023

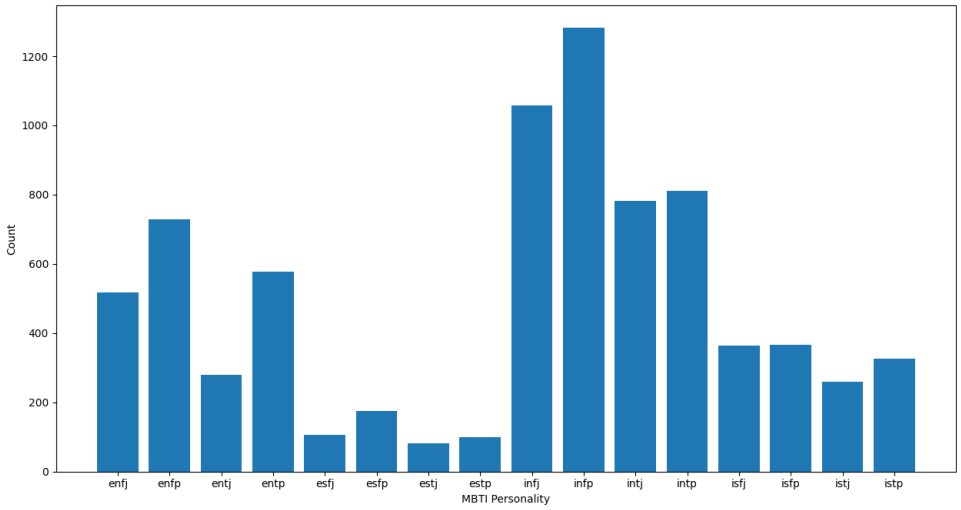
0.1 Wstęp

Korzystając ze zbioru danych MBTI Personality Type Twitter Dataset zawierającego jako cechę posty danego użytkownika na Twitterze, a jako etykietę jego typ osobowości według klasyfikacji Myers-Briggs, zastosowaliśmy rózne metody uczenia maszynowego, aby sprawdzić z jaką dokładnością będą w stanie klasyfikować osobowość użytkownika na podstawie jego postów. Klasyfikacja Myers-Briggs wyróżnia 16 osobowości, co jest też liczbą unikalnych etykiet w naszym zbiorze danych, zatem losując odpowiedź otrzymamy średnio dokładność wynoszącą 6.25%. Określenie osobowości danej osoby na podstawie kilku postów na Twitterze może wydawać się niemożliwe dla przeciętnej osoby, ale metody uczenia maszynowego często są w stanie znaleźć nieoczywiste wzorce, dlatego choć mało prawdopodobne jest osiągnięcie dokładności na poziomie 98%, ciekawe jest jaką dokładność uda się osiągnąć. Kod projektu znajduję się w repozytorium na GitHubie.



0.2 Wstępna analiza danych

Ze wstępnej analizy danych możemy zauważyć, że liczba elementów przypadająca każdej etykiecie nie jest równa - między najliczniejszą etykietą zawierającą 1282 elementów (INFJ), a najmniej liczną zawierającą jedynie 81 elementów (ESTJ) jest aż 1201 elementów różnicy. Jest to spowodowane wieloma czynnikami - liczba osób o danym typie osobowości w społeczeństwie nie jest równa oraz pewne typy osobowości mogą być bardziej lub mniej skołonne do korzystania z Twittera.



Ze względu na taką dysproporcje będziemy badać zarówno jak metody działają dla całego zbioru, jak i próbki, w której każda etykieta jest równie prawdopodobna. W swojej analizie postów z Twittera, Garg S. i Garg A., 2021, [GG21] użyli niezbalansowanych danych, argumentując, że idealne zbalansowanie danych doprowadziłoby albo do wielu powtórzeń, albo do znacznego ograniczenia zbioru danych.

0.3 Feature extraction

Gdy na wejściu mamy ciągły tekst, zanim będziemy mogli zastosować na nim metody uczenia maszynowego, musimy dokonać jego obróbki, m.in. usunąć zbędne znaki i słowa oraz znormalizować wielkość liter, oraz zamienić go z ciągłego tekstu w wektor cech. Istnieje wiele sposobów na zrobienie tego, dlatego zdecydowaliśmy się przetestować kilka najpopularniejszych i najbardziej obiecujących:

0.3.1 Bags of Words

W tej metodzie liczymy liczbę wystąpień każdego słowa. Jest to bardzo prosta w implementacji metoda, ale ma kilka wad - otrzymujemy bardzo duży wektor w którym większość wartości to 0, co na przeciętnym komputerze szybko doprowadza do braku pamięci RAM. Aby zmniejszyć wektor, możemy nie brać pod uwagę słów, które występują w mniej niż n dokumentach lub więcej niż

n dokumentach, dzięki czemu pozbywamy się cech, które i tak nie miałyby dużego wpływu na algorytm, ponieważ dla każdej próbki miałyby taką samą wartość.

0.3.2 Term Frequency-Inverse Document Frequency

W metodzie TF-IDF liczymy *term frequency* $= tf(t,d) = \frac{f_{t,d}}{\sum f_{t',d}}$, czyli jak często słowo t występuje w dokumencie d oraz *inverse dokument frequency* $= idf(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$, czyli logarytm z odwrotność procentu dokumentów, w którym dane słowo występuje. Mając to policzone, dla każdego słowa możemy policzyć:

$$tfidf(t,d,D) = tf(t,d) * idf(t,D)$$

Intuicja jest taka, że wysoką wagę mają słowa, które występują często w małej liczbie dokumentów, czyli są dobrym elementem wyróżniającym daną podgrupę.

0.3.3 Porównanie

Przetestowaliśmy różne metody uczenia maszynowego na wektorach uzyskanych metodą TF-IDF i Bags of Words, a wyniki prezentują się następująco:

Method	**Bags of words**	**TF-IDF**
Random Forest	43.5%	45.5%
SVM	31.6%	39.7%
Naive Bayes	25.0%	17.8%
Logistic Regression	29.9%	30.5%

W większości, nasze obserwacje są zgodne z pracą Garg S. i Garg A., 2021, [GG21] według której TF-IDF produkuję wektory na których metody uczenia maszynowego otrzymują lepsze wyniki niż na wektorach wyprodukowanych przez Bags of Words. Wyjątkiem jest Naiwny Bayes, który ma lepsze wyniki z metodą Bags of Words. Badaniem różnych metod wyciągania cech z tekstu zajęli się także Ahuja et. al, 2019, [Ahu+19] którzy również doszli do wniosku, że metoda TF-IDF produkuje najlepsze rezultaty.

0.4 Porównanie metod

0.4.1 Lasy Losowe

Przetestowaliśmy metodę Lasów Losowych na różnej ilości drzew, z maksymalną głębokością drzew ograniczoną do 20:

# Trees	Avg. Accuracy
100	24.5%
200	35.1%
500	34.2%
1000	33.9%

Jak widać, przy 200 drzewach metoda uzyskuje optymalne wyniki. Jeżeli nie ograniczymy maksymalnej głębokości drzew, tylko będziemy kontynuować dopóki wszystkie liście nie będą czyste, to metoda znacznie spowalnia, ale jesteśmy w stanie uzyskać średnią dokładność wynoszącą **45.5%**.

0.4.2 Maszyny Wektorów Nośnych

Dla ustalonych parametrów, zmieniając jedynie funkcje jądrowe dla SVM, zbadaliśmy jaką dokładność otrzymają:

Kernel	Avg. Accuracy
Linear	25.5%
Polynomial	17.2%
Sigmoid	23.2%

Dla wszystkich funkcji osiągnięta dokładność jest niska, co wynika z potrzeby ograniczenia liczby danych treningowych, ze względu na duże zużycie zasobów przez SVM, ale widać, że liniowa funkcja jądrowa osiąga najlepsze wyniki. Z tą wiedzą, możemy użyć SVM z liniową funkcją jądrową na całym zbiorze i zobaczyć jaką dokładność uda się uzyskać. Działając na całym zbiorze danych, SVM otrzymuje dokładność wynoszącą **39.7%**.

0.4.3 Naiwny Bayes

W naiwnym klasyfikatorze bayesowskim podstawowym parametrem, który można przetestować, jest założenie o funkcji wiarygodności. Zbadaliśmy następujące hipotezy:

Likelihood	Avg. Accuracy
Gaussian	21.5%
Multinomial	17.1%
Complement	25.8%
Categorical	16.3%

Najlepiej się sprawdził tzw. Complement Naive Bayes, który został zaproponowany przez [Ren+03] m.in. z myślą o niezbalansowanych zbiorach danych, tak jak mamy do czynienia w tym przypadku.

Metoda polega na użyciu dopełnień klas do nadania im wag w następujących krokach:

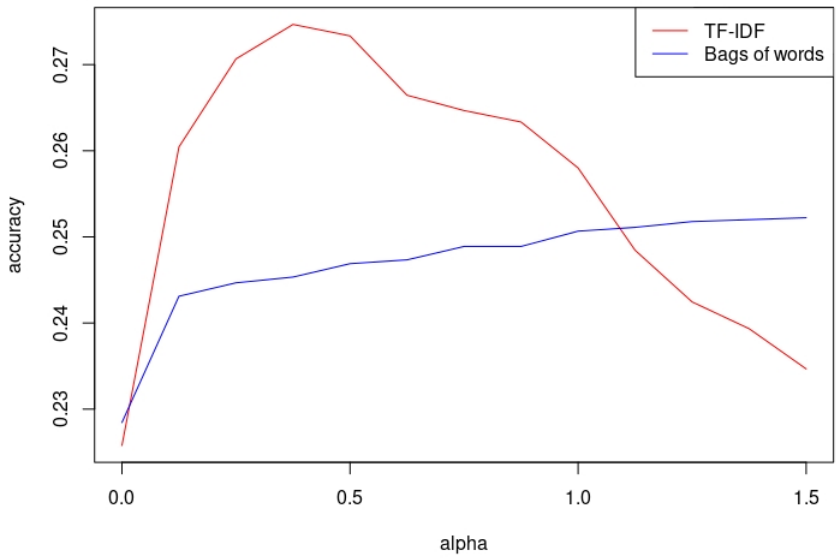
$$\hat{\theta}_{ci} = \frac{\alpha_i + \sum_{j:y_j \neq c} d_{ij}}{\alpha + \sum_{j:y_j \neq c} \sum_k d_{kj}}$$
$$w_{ci} = \log \hat{\theta}_{ci}$$
$$w_{ci} = \frac{w_{ci}}{\sum_j |w_{cj}|}$$

Następnie elementy są klasyfikowane zgodnie z zasadą

$$\hat{c} = \arg\min_c \sum_i t_i w_{ci}$$

tzn. przypisywana jest klasa o najgorzej dopasowanym dopełnieniu.

Drugim testowanym przez nas parametrem, już tylko na modelu CNB, był współczynnik wygładzenia Laplace'a. Co ciekawe, zupełnie odmienne wartości okazały się optymalne w zależności od metody przygotowania danych: 0.325 dla TF-IDF, 1.25 dla Bags of words.



0.4.4 Regresja Logistyczna

Regresja Logistyczna działała zdecydowanie najwolniej ze wszystkich testowanych przez nas metod. Mimo tego, nie osiągała zbyt dobrych wyników, mając średnią dokładność na poziomie **30.5%**.

0.4.5 Sieć Neuronowa

Przetestowaliśmy kilka modeli o różnych licznościach warstw pośrednich. Pomimo dużego narzutu czasowego, otrzymane wyniki się szczególnie nie wyróżniały:

Layers	Avg. Accuracy
(300, 300)	19.7%
(300)	23.8%
(100)	22.9%
(20, 20)	18.4%

0.5 Klasyfikacja binarna

Zamiast klasyfikować posty do jednej z 16 klas, możemy zawęzić naszą klasyfikację do pojedynczych składowych MBTI. Są nimi ekstrawersja-introwersja, poznanie-intuicja, myślenie-odczuwanie i osądzanie-obserwacja. Moreno et al., 2019, [Mor+19] którzy pracowali na zbiorze danych trochę mniejszym od tego na którym my pracowaliśmy, dokonali klasyfikacji osobowości patrząc na cechy, które są analogiczne do składowych części klasyfikacji Myers-Briggs, czyli osi E-I, N-S, T-F oraz P-J. Posty wektoryzowali za pomocą metody TFIDF i korzystali z SVM o liniowej funkcji jądrowej, Regresji Logistycznej oraz Lasów Losowych. Otrzymali następujące wyniki:

Trait	SVM	Logistic Regression	Random Forests
Extroverted	70%	55%	25%
Stable	65%	50%	32%
Agreeable	67%	49%	34%
Conscientous	63%	59%	27%
Open	73%	60%	36%

Udało im się osiągnąć dokładność lepszą niż zgadywanie, ale jak widać oscyluje ona w okolicach 70%, co potwierdza sugestię, że przewidywanie typu osobowości na podstawie postów na Twitterze jest trudne. Ciekawy jest słaby wynik Lasów Losowych, ale może on być spowodowany słabym doborem hiperparametrów. Garg S. i Garg A., 2021, [GG21] którzy dysponowali zbiorem danych znacznie większym od naszego otrzymali następujące wyniki:

Trait	SVM	Logistic Regression	Random Forests
Extroverted	79.61%	81.24%	99.96%
Agreeable	78.03%	81.68%	98.98%
Conscientous	81.79%	95.05%	99.99%
Open	85.32%	87.03%	99.88%

Rzutując nasze etykiety, aby brały pod uwagę tylko jedną ze składowych typu osobowości otrzymaliśmy następujące wyniki:

Method	E-I	N-S	T-F	P-J
Random Forest	72.6%	79.4%	70.8%	68.9%
SVM	71.3%	79.7%	69.8%	69.7%
Naive Bayes	66.7%	77.4%	61.1%	61.0%
Logistic Regression	68.3%	77.5%	68.4%	69.3%

Ponieważ nasz zbiór danych jest niezbalansowany, policzyliśmy *confusion matrix* dla każdego modelu, ponieważ możliwe jest, że zamiast nauczyć się przewidywać etykiety na podstawie cech, model nauczy się jedynie tego, która etykieta jest bardziej prawdopodobna. Pod tym względem najlepiej zachowuje się SVM, który mimo, że częściej przewiduje etykietę "I" niż rzeczywiście ona występuje, to regularnie próbuje zgadnąć etykietę "E":

SVM

		predicted value		
		E	**I**	**total**
actual value	**E**	175	346	521
	I	102	940	1042
	total	277	1286	

Lasy Losowe, które otrzymały dokładność podobną do SVM, mylą się na podobnej liczbie przykładów, ale prawie zawszę mylą się przewidując "I", gdy rzeczywistą etykietą było "E", podczas gdy SVM popełniał oba rodzaje błędów:

Random Forest

		predicted value		
		E	**I**	**total**
actual value	**E**	96	425	521
	I	3	1039	1042
	total	99	1464	

Multinomial Naive Bayes zrobił to, czego się obawialiśmy, czyli nauczył się tylko tego, która cecha jest bardziej prawdopodobna i na tej podstawie za każdym razem zgadywał "I":

Multinomial Naive Bayes

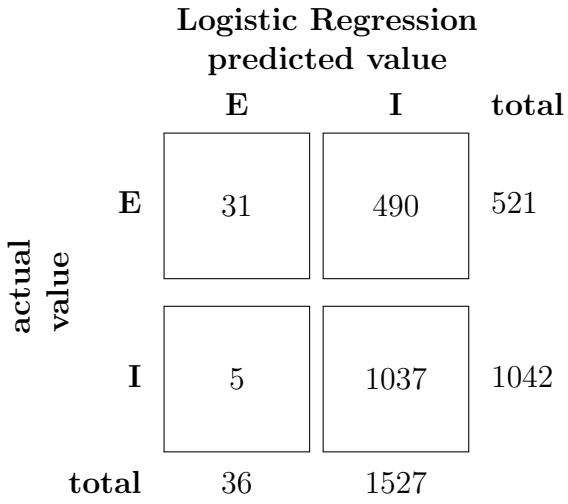
actual value \ predicted value	**E**	**I**	**total**
E	1	520	521
I	1	1041	1042
total	2	1561	

Complement Naive Bayes, który powinien lepiej działać na niezbalansowanych zbiorach, faktycznie próbuje zgadywać etykietę "E", ale mimo tego myli się podobnie często, co Multinomial Naive Bayes, jedynie w inny sposób:

Complement Naive Bayes

actual value \ predicted value	**E**	**I**	**total**
E	82	439	521
I	91	951	1042
total	173	1390	

Regresja Logistyczna, podobnie jak Multinomial Naive Bayes, w znacznej większości przypadków woli przewidywać etykietę "I", przewidując "E" jedynie kilka razy:



Nasze wyniki są średnio lepsze niż wyniki Moreno et al., 2019, [Mor+19] co jest najprawdopodobniej spowodowane tym, że dysponujemy większym zbiorem danych, ale nie udało się przekroczyć 80%. Garg S. i Garg A., 2021, [GG21] otrzymali znacznie lepsze wyniki, ale należy zaznaczyć, że mieli do dyspozycji znacznie większy zbiór danych od naszego. Zarówna Lasy Losowe, jak i Metoda Wektorów Nośnych sprawują się bardzo dobrze. Możemy też zauważyć, że niektóre cechy są znacznie łatwiejsze do wykrycia (N-S), a niektóre trudniejsze (T-F).

0.6 Równomierny zbiór danych

Próbkowanie zbioru danych w taki sposób, aby każda etykieta była równie prawdopodobna dało następująco rezultaty:

Method	Even dataset	Full dataset
Random Forest	45.0%	45.5%
SVM	18.5%	39.7%
Naive Bayes	11.2%	17.8%
Logistic Regression	14.2%	30.5%

Jak widać, otrzymane wyniki są znacznie gorsze, co wynika z tego, że wyrównanie zbioru danych daje nam zbiór danych o znacznie mniejszych rozmiarach. Ciekawe jest, że zmniejszenie zbioru danych nie miało istotnego wpływu na Lasy Losowe.

0.7 Inne zbiory danych

W celu zweryfikowania poprawności naszych modeli wykorzystaliśmy je na innym zbiorze danych, (MBTI) Myers-Briggs Personality Type Dataset, zawierającym jako cechy posty użytkowników na forum PersonalityCafe, a jako etykiety ich typ osobowości według klasyfikacji Myers-Briggs.

Method	New dataset	Original dataset
Random Forest	47.0%	45.5%
SVM	62.9%	39.7%
Naive Bayes	20.8%	17.8%
Logistic Regression	55.3%	30.5%

Na nowym zbiorze danych udało nam się w każdej metodzie uzyskać lepsze wyniki. Najlepszą metodą okazała się Metoda Wektorów Nośnych, która uzyskała dokładność **62.9%**.

0.8 Podsumowanie

Okazuje się, że za pomocą metod uczenia maszynowego jak najbardziej jesteśmy w stanie sklasyfikować typ osobowości danej osoby na podstawie jej postów na Twitterze. Na wykorzystywanym przez nas zbiorze danych udało się osiągnąć dokładność wynoszącą 45.5% za pomocą metody Lasów Losowych. Jest to dalekie od 100%, ale biorąc pod uwagę, że klasyfikacja miała 16 możliwych etykiet, czyli losowo przypisując etykiety mielibyśmy dokładność 6.25%, taki wynik można uznać za sukces i wskazuje on na to, że posty, które piszemy są skorelowane z naszym typem osobowości. Zgodnie z obserwacjami poczynionymi przez Garg S. i Garg A., 2021, [GG21] oraz Ahuja et. al, 2019, [Ahu+19] TF-IDF jest bardzo efektywną metodą wyciąganie cech z tekstu. Udało nam się uzyskać lepsze wyniki niż podobne prace zajmujące się mniejszymi zbiorami danych, ale gorsze niż podobne prace zajmujące się większymi zbiorami danych, co sugeruje, że aby wytrenować lepszy model należałoby przygotować większy zbiór danych.

Bibliography

[Ren+03] Jason D. M. Rennie et al. "Tackling the Poor Assumptions of Naive Bayes Text Classifiers". In: ICML'03 (2003), pp. 616–623. URL: https://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf.

[Ahu+19] Ravinder Ahuja et al. "The Impact of Features Extraction on the Sentiment Analysis". In: *Procedia Computer Science* 152 (2019). International Conference on Pervasive Computing Advances and Applications- PerCAA 2019, pp. 341–348. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.05.008. URL: https://www.sciencedirect.com/science/article/pii/S1877050919306593.

[Mor+19] Daniel Jaimes Moreno et al. "Prediction of Personality Traits in Twitter Users with Latent Features". In: Mar. 2019. DOI: 10.1109/CONIELECOMP.2019.8673242.

[GG21] Shruti Garg and Ashwani Garg. "Comparison of machine learning algorithms for content based personality resolution of tweets". In: *Social Sciences & Humanities Open* 4.1 (2021), p. 100178. ISSN: 2590-2911. DOI: https://doi.org/10.1016/j.ssaho.2021.100178. URL: https://www.sciencedirect.com/science/article/pii/S2590291121000747.