Wprowadzenie do Data Science i metod uczenia maszynowego **2020/2021**

Prowadzący: mgr inż. Rafał Woźniak Wtorek, 13:15

Szymon Gruda	239661	239661@edu.p.lodz.pl
Jan Karwowski	239671	239671@edu.p.lodz.pl
Michał Kidawa	239673	239673@edu.p.lodz.pl
Kamil Kowalewski	239676	239676@edu.p.lodz.pl

Zadanie 6.: Problem Set 6

Spis treści

1. Wprowadzenie

1.1. Principal Component Analysis

Na początku warto wspomnieć, że do badań zostały wykorzystane dwa parametry, pierwszym z nich jest liczba komponentów nazwana w bibliotece *sklearn* jako *n_components*. Są to procentowe wartości liczby kolumn jakie powinny pozostać po dokonaniu redukcji wielowymiarowości, zostały wykorzystane wartości z przedziału od 30% to 95%. Ze względu na drugi parametr wykorzystany w badaniach jakim jest *svd_solver* czyli rodzaj użytej metody SVD wartości procentowe zostały przemnożone przez liczbę kolumn lub wierszy w danym zbiorze. Wybór kolumn lub wierszy jest podyktowany logiką implementacji tej metody i brane jest pod uwagę minimum z liczby kolumn i wierszy i mnożone przez wybrany procent. Co do samych wersji *svd_solver* to zostały wykorzystane wszystkie dostępne warianty czyli *auto*, *full*, *arpack*, *randomized*. Stąd w kolumnie *parametry metody* jest np *pca_auto_185* co oznacza, że została wykorzystana metoda PCA z SVD ustawionym na *auto* oraz liczbę kolumn, która ma zostać na 185.

1.2. Singular Value Decomposition

Tak jak w przypadku metody *Principal Component Analysis* podczas eksperymentów, zmianie ulegały wartości dwóch parametrów. Pierwszym z nich jest liczba komponentów - *n_components*, która przyjmowała wartość liczby kolumn pomnożonej przez liczby z przedziału od 0.2 do 0.9 (które reprezentują procenty nominalnej liczby kolumn). Drugim z badanych parametrów był rodzaj algorytmu wykorzystywany przy redukcji wielowymiarowości. Do dyspozycji były 2 warianty - *arpack* oraz *randomized*. Analogicznie do *Principal Component Analysis* znajduje to odzwierciedlenie w wartościach zapisanych w kolumnie *parametry metody* - *svd_arpack_123* oznacza wykorzystanie metody SVD z algorytmem *arpack* i liczbą kolumn równą 123.

1.3. Analiza wariancji

Metoda selekcji cech (a tym samym redukcji wymiarów) oparta o analizę wariancji sprowadza się do oceny „ważności" cech, na podstawie prostego kryterium: im większa wariancja danej cechy w całym zbiorze treningowym i im mniejsza wariancja tej cechy w ramach każdej z klas, tym lepsza jest ta cecha. Tak więc każda cecha ma przyporządkowaną wagę, zgodną ze wzorem:

$$w_j = \frac{\sigma_j}{\frac{1}{K}\sum_{k=1}^{K}\sigma_j^k} \tag{1}$$

gdzie w_j to waga j-tej cechy, σ_j to wariancja j-tej cechy we wszystkich próbkach uczących, a σ_j^k to wariancja j-tej cechy w ramach próbek z klasy k (wszystkich klas jest K). Tak więc czym większa wariancja w ramach całego zbioru i czym mniejsza wariancja w ramach poszczególnych klas, tym cecha jest lepsza. Dodatkowo, jeżeli wartości w liczebniku i mianowniku są równe 0, to cecha taka jest uznawana za całkowicie nieprzydatną i ma przypisaną

możliwie małą wagę. Po wycenie wszystkich cech można przystąpić do właściwiej selekcji, która sprowadza się do posortowania cech od najlepszej (z największą wagą) do najgorszej i wybraniu n (czyli ile się chce) najlepszych cech.

1.4. Correlation-based Feature Selection

Jednym z najbardziej zwięzłych, a przy tym najlepiej opisującym działanie metody CFS jest zdanie: *"Dobre podzbiory cech zawierają cechy silnie skorelowane z klasyfikacją, ale nieskorelowane między sobą."*. Metoda ta oblicza metrykę "merit" dla każdego podzbioru, a następnie na podstawie tych wartości tworzy najlepszy (według metody CFS) podzbiór dla danego zbioru. Na skutek liczby obliczeń, które musi wykonać program realizujący ten algorytm, dostępne implementacje są mało wydajne, przez co znalezienie podzbioru klas, składającego się z więcej niż 20 elementów, jeżeli weźmie się pod uwagę złożone zbiory danych, trwa bardzo długo.

1.5. Opis zbiorów danych

1.5.1. Letters

Jest to zbiór [1] zawierający wyekstrachowane cechy z nagrań wypowiadanych liter alfabetu. Badane osoby w liczbie 150, dwukrotnie wypowiadały cały alfabel stąd mamy 52 nagrania dla każdej osoby a następnie z nich zostały wyekstrachowane cechy przez autorów tego zbioru danych.

1.5.2. Numerals

Jest to zbiór [2] zawierający wyekstrachowane cechy z odręcznie pisanych cyfr od *0* do *9* ze zbioru holenderskich map użyteczności publicznej.

1.5.3. Documents

Jest to zbiór [3] zawierający wyekstrachowane cechy ze zbioru 1080 dokumentów opisów biznesowych brazylijskich firm. Został na słowach w dokumentach przeprowadzony proces usuwania przyimków i słowa na wektor cech został zamieniony poprzez zastowanie miary jaką jest częstotliwość słowa w dokumencie.

2. Wyniki

2.1. Principal Component Analysis

2.1.1. Zbiór letters

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.92	random_forest	0.94
pca_auto_185	knn	0.919	random_forest	0.911
pca_auto_216	knn	0.917	random_forest	0.899
pca_auto_247	knn	0.919	random_forest	0.903
pca_auto_278	knn	0.918	random_forest	0.899
pca_auto_308	knn	0.919	random_forest	0.895
pca_auto_339	knn	0.919	random_forest	0.886
pca_auto_370	knn	0.92	random_forest	0.885
pca_auto_401	knn	0.922	random_forest	0.881
pca_auto_432	knn	0.92	random_forest	0.885
pca_auto_463	knn	0.922	random_forest	0.885
pca_auto_494	knn	0.921	random_forest	0.886
pca_auto_524	knn	0.92	random_forest	0.874
pca_auto_555	knn	0.92	random_forest	0.885
pca_auto_586	knn	0.92	random_forest	0.869
pca_full_185	knn	0.918	random_forest	0.911
pca_full_216	knn	0.918	random_forest	0.896
pca_full_247	knn	0.919	random_forest	0.892
pca_full_278	knn	0.917	random_forest	0.899
pca_full_308	knn	0.918	random_forest	0.894
pca_full_339	knn	0.919	random_forest	0.897
pca_full_370	knn	0.922	random_forest	0.898
pca_full_401	knn	0.922	random_forest	0.874
pca_full_432	knn	0.922	random_forest	0.884
pca_full_463	knn	0.92	random_forest	0.883
pca_full_494	knn	0.921	random_forest	0.886
pca_full_524	knn	0.92	random_forest	0.874
pca_full_555	knn	0.92	random_forest	0.885
pca_full_586	knn	0.92	random_forest	0.869

Tabela 1. Zbiór letters cz.1

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
pca_arpack_185	knn	0.918	random_forest	0.911
pca_arpack_216	knn	0.918	random_forest	0.896
pca_arpack_247	knn	0.919	random_forest	0.892
pca_arpack_278	knn	0.917	random_forest	0.899
pca_arpack_308	knn	0.918	random_forest	0.894
pca_arpack_339	knn	0.919	random_forest	0.897
pca_arpack_370	knn	0.922	random_forest	0.898
pca_arpack_401	knn	0.922	random_forest	0.874
pca_arpack_432	knn	0.922	random_forest	0.884
pca_arpack_463	knn	0.92	random_forest	0.883
pca_arpack_494	knn	0.921	random_forest	0.886
pca_arpack_524	knn	0.92	random_forest	0.874
pca_arpack_555	knn	0.92	random_forest	0.885
pca_arpack_586	knn	0.92	random_forest	0.869
pca_randomized_185	knn	0.919	random_forest	0.911
pca_randomized_216	knn	0.917	random_forest	0.899
pca_randomized_247	knn	0.919	random_forest	0.903
pca_randomized_278	knn	0.918	random_forest	0.899
pca_randomized_308	knn	0.919	random_forest	0.895
pca_randomized_339	knn	0.919	random_forest	0.886
pca_randomized_370	knn	0.92	random_forest	0.885
pca_randomized_401	knn	0.922	random_forest	0.881
pca_randomized_432	knn	0.92	random_forest	0.885
pca_randomized_463	knn	0.922	random_forest	0.885
pca_randomized_494	knn	0.92	random_forest	0.881
pca_randomized_524	knn	0.92	random_forest	0.883
pca_randomized_555	knn	0.92	random_forest	0.883
pca_randomized_586	knn	0.92	random_forest	0.876

Tabela 2. Zbiór letters cz.2

2.1.2. Zbiór numerals

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.952	random_forest	0.985
pca_auto_195	knn	0.952	random_forest	0.962
pca_auto_227	knn	0.952	random_forest	0.963
pca_auto_260	knn	0.952	random_forest	0.962
pca_auto_292	knn	0.952	random_forest	0.962
pca_auto_324	knn	0.952	random_forest	0.958
pca_auto_357	knn	0.952	random_forest	0.963
pca_auto_389	knn	0.952	random_forest	0.953
pca_auto_422	knn	0.952	random_forest	0.955
pca_auto_454	knn	0.952	random_forest	0.958
pca_auto_487	knn	0.952	random_forest	0.955
pca_auto_519	knn	0.952	random_forest	0.955
pca_auto_552	knn	0.952	random_forest	0.963
pca_auto_584	knn	0.952	random_forest	0.957
pca_auto_617	knn	0.952	random_forest	0.95
pca_full_195	knn	0.952	random_forest	0.963
pca_full_227	knn	0.952	random_forest	0.958
pca_full_260	knn	0.952	random_forest	0.957
pca_full_292	knn	0.952	random_forest	0.957
pca_full_324	knn	0.952	random_forest	0.963
pca_full_357	knn	0.952	random_forest	0.957
pca_full_389	knn	0.952	random_forest	0.958
pca_full_422	knn	0.952	random_forest	0.96
pca_full_454	knn	0.952	random_forest	0.952
pca_full_487	knn	0.952	random_forest	0.952
pca_full_519	knn	0.952	random_forest	0.957
pca_full_552	knn	0.952	random_forest	0.963
pca_full_584	knn	0.952	random_forest	0.957
pca_full_617	knn	0.952	random_forest	0.95

Tabela 3. Zbiór numerals cz.1

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
pca_arpack_195	knn	0.952	random_forest	0.963
pca_arpack_227	knn	0.952	random_forest	0.958
pca_arpack_260	knn	0.952	random_forest	0.957
pca_arpack_292	knn	0.952	random_forest	0.957
pca_arpack_324	knn	0.952	random_forest	0.963
pca_arpack_357	knn	0.952	random_forest	0.957
pca_arpack_389	knn	0.952	random_forest	0.958
pca_arpack_422	knn	0.952	random_forest	0.96
pca_arpack_454	knn	0.952	random_forest	0.952
pca_arpack_487	knn	0.952	random_forest	0.952
pca_arpack_519	knn	0.952	random_forest	0.957
pca_arpack_552	knn	0.952	random_forest	0.963
pca_arpack_584	knn	0.952	random_forest	0.96
pca_arpack_617	knn	0.952	random_forest	0.942
pca_randomized_195	knn	0.952	random_forest	0.962
pca_randomized_227	knn	0.952	random_forest	0.963
pca_randomized_260	knn	0.952	random_forest	0.962
pca_randomized_292	knn	0.952	random_forest	0.962
pca_randomized_324	knn	0.952	random_forest	0.958
pca_randomized_357	knn	0.952	random_forest	0.963
pca_randomized_389	knn	0.952	random_forest	0.953
pca_randomized_422	knn	0.952	random_forest	0.955
pca_randomized_454	knn	0.952	random_forest	0.958
pca_randomized_487	knn	0.952	random_forest	0.955
pca_randomized_519	knn	0.952	random_forest	0.955
pca_randomized_552	knn	0.952	random_forest	0.963
pca_randomized_584	knn	0.952	random_forest	0.953
pca_randomized_617	knn	0.952	random_forest	0.952

Tabela 4. Zbiór numerals cz.2

2.1.3. Zbiór documents

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.867	random_forest	0.92
pca_auto_227	knn	0.852	random_forest	0.892
pca_auto_265	knn	0.84	random_forest	0.895
pca_auto_302	knn	0.843	random_forest	0.895
pca_auto_340	knn	0.843	random_forest	0.898
pca_auto_378	knn	0.836	random_forest	0.91
pca_auto_416	knn	0.84	random_forest	0.907
pca_auto_454	knn	0.83	random_forest	0.889
pca_auto_491	knn	0.827	random_forest	0.904
pca_auto_529	knn	0.846	random_forest	0.864
pca_auto_567	knn	0.852	random_forest	0.877
pca_auto_605	knn	0.873	random_forest	0.889
pca_auto_643	knn	0.864	random_forest	0.898
pca_auto_680	knn	0.867	random_forest	0.88
pca_auto_718	knn	0.867	random_forest	0.886
pca_full_227	knn	0.849	random_forest	0.898
pca_full_265	knn	0.843	random_forest	0.904
pca_full_302	knn	0.846	random_forest	0.892
pca_full_340	knn	0.843	random_forest	0.898
pca_full_378	knn	0.836	random_forest	0.904
pca_full_416	knn	0.843	random_forest	0.901
pca_full_454	knn	0.833	random_forest	0.904
pca_full_491	knn	0.83	random_forest	0.898
pca_full_529	knn	0.867	random_forest	0.864
pca_full_567	knn	0.867	random_forest	0.877
pca_full_605	knn	0.873	random_forest	0.889
pca_full_643	knn	0.864	random_forest	0.898
pca_full_680	knn	0.867	random_forest	0.88
pca_full_718	knn	0.867	random_forest	0.886

Tabela 5. Zbiór documents cz.1

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
pca_arpack_227	knn	0.849	random_forest	0.898
pca_arpack_265	knn	0.843	random_forest	0.904
pca_arpack_302	knn	0.846	random_forest	0.892
pca_arpack_340	knn	0.843	random_forest	0.898
pca_arpack_378	knn	0.836	random_forest	0.904
pca_arpack_416	knn	0.843	random_forest	0.901
pca_arpack_454	knn	0.833	random_forest	0.904
pca_arpack_491	knn	0.83	random_forest	0.898
pca_arpack_529	knn	0.833	random_forest	0.864
pca_arpack_567	knn	0.843	random_forest	0.877
pca_arpack_605	knn	0.849	random_forest	0.889
pca_arpack_643	knn	0.849	random_forest	0.898
pca_arpack_680	knn	0.849	random_forest	0.88
pca_arpack_718	knn	0.846	random_forest	0.886
pca_randomized_227	knn	0.852	random_forest	0.892
pca_randomized_265	knn	0.84	random_forest	0.895
pca_randomized_302	knn	0.843	random_forest	0.895
pca_randomized_340	knn	0.843	random_forest	0.898
pca_randomized_378	knn	0.836	random_forest	0.91
pca_randomized_416	knn	0.84	random_forest	0.907
pca_randomized_454	knn	0.83	random_forest	0.889
pca_randomized_491	knn	0.827	random_forest	0.904
pca_randomized_529	knn	0.846	random_forest	0.864
pca_randomized_567	knn	0.852	random_forest	0.877
pca_randomized_605	knn	0.846	random_forest	0.889
pca_randomized_643	knn	0.83	random_forest	0.898
pca_randomized_680	knn	0.84	random_forest	0.88
pca_randomized_718	knn	0.855	random_forest	0.886

Tabela 6. Zbiór documents cz.2

2.2. Singular Value Decomposition

2.2.1. Zbiór letters

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.92	random_forest	0.94
svd_arpack_123	knn	0.92	random_forest	0.91
svd_arpack_185	knn	0.919	random_forest	0.903
svd_arpack_246	knn	0.919	random_forest	0.899
svd_arpack_308	knn	0.918	random_forest	0.892
svd_arpack_370	knn	0.922	random_forest	0.889
svd_arpack_431	knn	0.922	random_forest	0.888
svd_arpack_493	knn	0.921	random_forest	0.874
svd_arpack_555	knn	0.92	random_forest	0.881
svd_randomized_123	knn	0.92	random_forest	0.915
svd_randomized_185	knn	0.919	random_forest	0.912
svd_randomized_246	knn	0.917	random_forest	0.894
svd_randomized_308	knn	0.918	random_forest	0.903
svd_randomized_370	knn	0.92	random_forest	0.894
svd_randomized_431	knn	0.921	random_forest	0.871
svd_randomized_493	knn	0.921	random_forest	0.887
svd_randomized_555	knn	0.92	random_forest	0.876

Tabela 7. Wyniki klasyfikacji dla zbioru letters

2.2.2. Zbiór numerals

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.952	random_forest	0.985
svd_arpack_129	knn	0.952	random_forest	0.97
svd_arpack_194	knn	0.952	random_forest	0.967
svd_arpack_259	knn	0.952	random_forest	0.965
svd_arpack_324	knn	0.952	random_forest	0.972
svd_arpack_389	knn	0.952	random_forest	0.965
svd_arpack_454	knn	0.952	random_forest	0.963
svd_arpack_519	knn	0.952	random_forest	0.955
svd_arpack_584	knn	0.952	random_forest	0.957
svd_randomized_129	knn	0.952	random_forest	0.967
svd_randomized_194	knn	0.952	random_forest	0.965
svd_randomized_259	knn	0.952	random_forest	0.97
svd_randomized_324	knn	0.952	random_forest	0.96
svd_randomized_389	knn	0.952	random_forest	0.965
svd_randomized_454	knn	0.952	random_forest	0.952
svd_randomized_519	knn	0.952	random_forest	0.958
svd_randomized_584	knn	0.952	random_forest	0.955

Tabela 8. Wyniki klasyfikacji dla zbioru numerals

2.2.3. Zbiór documents

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.858	random_forest	0.92
svd_arpack_151	knn	0.852	random_forest	0.923
svd_arpack_226	knn	0.849	random_forest	0.898
svd_arpack_302	knn	0.846	random_forest	0.904
svd_arpack_378	knn	0.836	random_forest	0.901
svd_arpack_453	knn	0.836	random_forest	0.877
svd_arpack_529	knn	0.836	random_forest	0.883
svd_arpack_604	knn	0.84	random_forest	0.883
svd_arpack_680	knn	0.858	random_forest	0.898
svd_randomized_151	knn	0.852	random_forest	0.907
svd_randomized_226	knn	0.846	random_forest	0.907
svd_randomized_302	knn	0.846	random_forest	0.892
svd_randomized_378	knn	0.836	random_forest	0.914
svd_randomized_453	knn	0.836	random_forest	0.898
svd_randomized_529	knn	0.849	random_forest	0.883
svd_randomized_604	knn	0.833	random_forest	0.883
svd_randomized_680	knn	0.858	random_forest	0.898

Tabela 9. Wyniki klasyfikacji dla zbioru documents

2.3. Analiza wariancji

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.92	random_forest	0.94
va_0.01_6	knn	0.266	random_forest	0.257
va_0.02_12	knn	0.52	random_forest	0.561
va_0.03_18	knn	0.648	random_forest	0.69
va_0.04_24	knn	0.693	random_forest	0.737
va_0.05_30	knn	0.706	random_forest	0.76
va_0.08_49	knn	0.766	random_forest	0.824
va_0.1_61	knn	0.807	random_forest	0.852
va_0.2_123	knn	0.847	random_forest	0.895
va_0.3_185	knn	0.906	random_forest	0.93
va_0.4_246	knn	0.918	random_forest	0.935
va_0.5_308	knn	0.913	random_forest	0.934
va_0.6_370	knn	0.914	random_forest	0.94
va_0.7_431	knn	0.925	random_forest	0.939
va_0.8_493	knn	0.926	random_forest	0.942
va_0.9_555	knn	0.923	random_forest	0.942

Tabela 10. Wyniki klasyfikacji po analizie wariancji dla zbioru „letters”

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.952	random_forest	0.985
va_0.01_6	knn	0.738	random_forest	0.917
va_0.02_12	knn	0.85	random_forest	0.95
va_0.03_19	knn	0.835	random_forest	0.972
va_0.04_25	knn	0.83	random_forest	0.98
va_0.05_32	knn	0.84	random_forest	0.982
va_0.08_51	knn	0.907	random_forest	0.985
va_0.1_64	knn	0.92	random_forest	0.987
va_0.2_129	knn	0.937	random_forest	0.987
va_0.3_194	knn	0.938	random_forest	0.988
va_0.4_259	knn	0.947	random_forest	0.978
va_0.5_324	knn	0.948	random_forest	0.982
va_0.6_389	knn	0.948	random_forest	0.983
va_0.7_454	knn	0.948	random_forest	0.982
va_0.8_519	knn	0.948	random_forest	0.985
va_0.9_584	knn	0.948	random_forest	0.983

Tabela 11. Wyniki klasyfikacji po analizie wariancji dla zbioru „numerals”

Parametry metody	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.867	random_forest	0.92
va_0.01_8	knn	0.444	random_forest	0.469
va_0.02_17	knn	0.722	random_forest	0.741
va_0.03_25	knn	0.738	random_forest	0.809
va_0.04_34	knn	0.79	random_forest	0.824
va_0.05_42	knn	0.79	random_forest	0.824
va_0.08_68	knn	0.846	random_forest	0.889
va_0.1_85	knn	0.867	random_forest	0.898
va_0.2_171	knn	0.873	random_forest	0.886
va_0.3_256	knn	0.861	random_forest	0.907
va_0.4_342	knn	0.873	random_forest	0.904
va_0.5_428	knn	0.873	random_forest	0.904
va_0.6_513	knn	0.877	random_forest	0.914
va_0.7_599	knn	0.88	random_forest	0.904
va_0.8_684	knn	0.873	random_forest	0.923
va_0.9_770	knn	0.867	random_forest	0.926

Tabela 12. Wyniki klasyfikacji po analizie wariancji dla zbioru „documents”

2.4. Correlation-based Feature Selection

2.4.1. Zbiór letters

Liczba klas w zbiorze	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.92	random_forest	0.94
1	knn	0.106	random_forest	0.096
2	knn	0.165	random_forest	0.167
3	knn	0.279	random_forest	0.274
4	knn	0.328	random_forest	0.322
5	knn	0.407	random_forest	0.396
6	knn	0.474	random_forest	0.519
7	knn	0.48	random_forest	0.516
8	knn	0.513	random_forest	0.575
9	knn	0.538	random_forest	0.585
10	knn	0.571	random_forest	0.613

Tabela 13. Tabela CFS dla zbioru letters

2.4.2. Zbiór numerals

Liczba klas w zbiorze	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.952	random_forest	0.985
1	knn	0.345	random_forest	0.357
2	knn	0.378	random_forest	0.338
3	knn	0.467	random_forest	0.433
4	knn	0.607	random_forest	0.593
5	knn	0.708	random_forest	0.672
6	knn	0.75	random_forest	0.743
7	knn	0.753	random_forest	0.748
8	knn	0.78	random_forest	0.803
9	knn	0.813	random_forest	0.848
10	knn	0.797	random_forest	0.857
11	knn	0.803	random_forest	0.87
12	knn	0.832	random_forest	0.907
13	knn	0.852	random_forest	0.902
14	knn	0.857	random_forest	0.91
15	knn	0.867	random_forest	0.912
16	knn	0.883	random_forest	0.922
17	knn	0.882	random_forest	0.92
18	knn	0.883	random_forest	0.92
19	knn	0.877	random_forest	0.927
20	knn	0.898	random_forest	0.935

Tabela 14. Tabela CFS dla zbioru numerals

2.4.3. Zbiór documents

Liczba klas w zbiorze	Klasyfikator	Accuracy	Klasyfikator	Accuracy
original	knn	0.867	random_forest	0.92
1	knn	0.213	random_forest	0.213
2	knn	0.306	random_forest	0.306
3	knn	0.38	random_forest	0.38
4	knn	0.451	random_forest	0.451
5	knn	0.522	random_forest	0.522
6	knn	0.568	random_forest	0.568
7	knn	0.562	random_forest	0.602
8	knn	0.577	random_forest	0.617
9	knn	0.633	random_forest	0.627
10	knn	0.688	random_forest	0.682
11	knn	0.716	random_forest	0.71
12	knn	0.744	random_forest	0.744
13	knn	0.725	random_forest	0.744
14	knn	0.731	random_forest	0.753
15	knn	0.738	random_forest	0.765
16	knn	0.701	random_forest	0.744
17	knn	0.701	random_forest	0.744
18	knn	0.688	random_forest	0.75
19	knn	0.71	random_forest	0.765
20	knn	0.728	random_forest	0.784

Tabela 15. Tabela CFS dla zbioru documents

3. Dyskusja

3.1. Principal Component Analysis

Patrząc na wyniki dla pierwszego zbióru danych [1] można zauważyć, że dla klasyfikatora KNN wartości *Accuracy* bez względu na parametry metody są praktycznie identyczne, różnice między najbardziej skrajnymi wynika to maksymalnie *0.01*. Co więcej wartości te są praktycznie równe z wynikiem uzyskanym dla oryginalnego zbioru danych. Przechodząc do klasyfikatora *random forest* można zauważyć, że wersja SVD nie wpływa jakoś szczególnie na wyniki, gdyż zazwyczaj są to różnice na poziomie *0.01* dla tych samych wartości *n_components*. Ciekawą obserwacją jest to, że dla najmniejszej wartości *n_components* równej *185* co jest równe *30%* uzyskano najlepszy wynik. Jest on o *0.03* gorszy od wyniku uzyskanego na oryginalnym zbiorze danych. Przy większej wartości *n_components* wynik *Accuracy* pogarsza się, w najgorzym przypadku był to spadek o *0.05* co i tak jest absolutnie akceptowalnym wynikiem.

Patrząc na wyniki dla drugiego zbióru danych [2] można zauważyć, że dla klasyfikatora KNN wartości *Accuracy* jest identyczna dla oryginalnego

zbioru oraz wszystkich kombinacji użycia metody PCA, *Accuracy* jest równe *0.952*. Przechodząc do klasyfikatora *random forest* można zauważyć, że najlepszy wynik uzyskany po zostasowaniu metody PCA jest o około *0.02* gorszy od wyniku dla oryginalnego zbioru. Kolejną tendencją jest to, że poszczególne wyniki*Accuracy* po zastosowaniu metody PCA różnią się nieznacznie około *0.02* i najlepszy wynik udało się uzyskać dla najmniejszej liczby *n_components*.

Patrząc na wyniki dla trzeciego zbióru danych [3] można zauważyć, że dla klasyfikatora KNN wartości *Accuracy* były najlepsze dla *svd_solver* w wersji *full* natomiast najgorsze dla *arpack*, gdzie najgorsze można rozumieć poprzez różnice w wynikach na poziomie *0.03*. Co więcej w przeciwieństwie do poprzednich zbiorów tutaj zmiejszanie wartości *n_components* wpływało na pogorszenie wyników, w połowie przypadków dla udało się uzyskać dla procentowej wartości *95%* oraz *90%* wynik identyczny jak w oryginalnym zbiorze. Przechodząc do klasyfikatora *random forest* można ponowie zauważyć, że wybór *svd_solver* nie wpływa znacząco na uzyskiwane wyniki oraz dla różnych wartości *n_components* wyniki są niemal identyczne.

Co warto wspomnieć w przypadku wybranych zbiorów danych wszystkie z nich miały rozmiar większy niz *500x500* co skutkowało tym, że metoda *auto* przełączała się na *randomized* aby poprawić wydajność.

3.2. Singular Value Decomposition

Analizując otrzymane wyniki można zauważyć, iż największa różnica accuracy pomiędzy oryginalnym zbiorem danych, a zredukowanym wynosi zaledwie 6.4 punkta procentowego. Dzieje się to dla zbioru letters w przypadku klasyfikatora *random forest* i SVD przy algorytmie *randomized* z liczbą kolumn wynoszącą *555*.

Obserwując czas działania programu i śledząc wykonywane kroki można było zauważyć, że redukcja wykonywana z algorytmem *arpack* zajmuje więcej czasu dla każdego zbioru danych.

W przypadku zbioru pierwszego zbioru danych [1] wyniki otrzymane przez klasyfikator *KNN* nie wykazują dużych różnic w zależności od liczby kolumn oraz stosowanego algorytmu. Dla *random forest* wraz ze zwiększaniem liczby kolumn malała wartość accuracy. Lepsze wyniki przy klasyfikacji za pomocą *random forest* osiągnął algorytm *arpack*.

Wyniki klasyfikacji dla zbioru [2] przedstawione w tabeli 8 prezentują identyczne wartości dla klasyfikatora *KNN* niezależnie od wykorzystywanego algorytmu przy SVD. Wszystkie wartości accuracy otrzymane przy tym zbiorze danych osiągają ponad 95% Można zatem wnioskować, że większość cech obecnych w przypadku tego zbioru danych jest zbędnych (na płaszczyźnie klasyfikacji).

Zbiór [3] pozwala zaobserwować wyniki podobne do tych otrzymanych w przypadku pierwszego zbioru danych. Tutaj również wraz ze zwiększeniem liczby kolumn danych wyjściowych maleje wartość accuracy, a zmiana algorytmu nie wpływa praktycznie wcale na otrzymane wyniki.

3.3. Analiza wariancji

Tabele 10, 11, 12 prezentują dokładność klasyfikacji po selekcji cech opartej o analizę wariancji. W pierwszej kolumnie każdej tabeli widzimy skrót zawierający dwie liczby - pierwsza z nich oznacza procent całkowitej liczby cech, a druga bezwzględną liczbę cech, zgodną z tym procentem dla wybranego zbioru danych. Skrót „original" oznacza zbiór przed redukcją wymiarów (wszystkie cechy obecne). Jak widać, we wszystkich trzech zbiorach udało się już przy około 30% osiągnąć dokładność klasyfikacji równą lub różniącą się o zaledwie kilka procent od tej, którą otrzymano dla pełnego zbioru danych. Zdecydowana większość cech jest więc zbędna. Warto wspomnieć, że nie mamy tutaj do czynienia z tworzeniem nowych cech na podstawie już istniejących - nadmiarowe cechy są po prostu usuwane i nie ma po nich żadnego śladu. Kluczowym obszarem, na którym zmienia się znacząco dokładność klasyfikacji jest pierwsze 10% liczby cech, a więc w przypadku analizowanych zbiorów, pierwsze kilkadziesiąt, a nawet kilkanaście.

Zdecydowanie najwięcej cech można było zredukować w przypadku zbioru „numerals", gdzie już pierwszych 6 cech (czyli zaledwie 1%), pozwoliło osiągnąć, z wykorzystaniem lasu losowego, 90% dokładności klasyfikacji. Przy 10% cech, wynik jest zdecydowanie bliski najlepszemu możliwemu. Zbiór „documents", choć trudniejszy do klasyfikacji, co widać po niższych wartościach dokładności dla obu klasyfikatorów, również udało się stosunkowo dobrze zredukować - przy 10% cech wynik jest również zbliżony do najlepszego. Najtrudniej było zredukować zbiór „letters" - przy 1% liczby cech dokładność jest rzędu 0.25, zatem zdecydowanie niewystarczająca. 5% daje już natomiast znacznie lepsze wyniki, bo ponad 0.7. Jednakże, aby osiągnąć wynik zbliżony do najlepszego (różniący się o kilka procent), trzeba już około 30% wszystkich cech, a więc kilka razy więcej, niż dla pozostałych zbiorów. Nasuwają się więc wnioski związane z wpływem charakterystyki zbioru na wyniki klasyfikacji po redukcji wymiarów. Czym bardziej abstrakcyjne i trudniejsze dane reprezentuje zbiór, tym więcej cech wymaga taki zbiór do poprawnej klasyfikacji. W przypadku przeprowadzonych eksperymentów, zbiór „letters" było najtrudniej zredukować - reprezentuje on natomiast nagrania ludzkiej mowy, która jest stosunkowo trudna w interpretacji przez maszynę. Najłatwiej udało się natomiast ze zbiorem „numerals", który reprezentuje nieskomplikowane dane graficzne - kilka podstawowych kształtów, które składają się na 10 cyfr. Faktem jest też, że dla tego drugiego wyekstrahowano po prostu zdecydowanie za dużo cech.

Warto jeszcze zastanowić się chwilę nad różnicą w wynikach dla różnych klasyfikatorów. Tak więc można zauważyć, że las losowy (*random forest*) radzi sobie w ogóle lepiej z klasyfikacją badanych zbiorów danych. Prawie wszystkie dokładności klasyfikacji są dla niego wyższe. Potrzebuje również mniej cech, aby móc dobrze klasyfikować. W przeciwieństwie do niego, znacznie mniej złożony KNN, potrzebuje więcej cech aby osiągnąć możliwie dobre wyniki. Widać to zwłaszcza na przykładzie zbioru danych „numerals". Po drugie, z przeprowadzonych eksperymentów wynika, że las losowy (z dokładnością do części setnych wartości miary *accuracy*) zazwyczaj osiąga lepsze wyniki wraz ze wzrostem liczby cech. W przypadku klasyfikatora KNN, war-

tości te bardziej się wahają i zdarzyło się nawet, tak jak w przypadku zbioru danych „documents", że najlepsza jakość klasyfikacji jest wyższa o ponad 2% dla zredukowanego zbioru danych, niż dla pełnego zbioru danych. Jest to zgodne z zasadą działania tego klasyfikatora, dla której decydujące znaczenie może mieć obecność pojedynczych próbek i wartości w zbiorze uczącym.

3.4. Correlation-based Feature Selection

Niestety, dla zbioru [1] nie udało się w akceptowalnym okresie czasu znaleźć optymalnej liczby elementów w podzbiorze klas, która nie wpłynęłaby na dokładność klasyfikacji. Jednakże warto zauważyć że dla liczby *10* wartości dokładności klasyfikacji to kolejno *0,571* i *0,613*. Jeżeli weźmiemy również szybkość wzrostu tych wartości dla kolejnych n to pozwala to przypuszczać, że przy większej liczbie czasu i zasobów, możliwe stałoby się wyznaczenie takiego podzbioru, dla którego klasyfikator klasyfikowałby równie dobrze, co dla całego zbioru.

Dla zbioru [2] udało się uzyskać wysoką dokładność klasyfikacji już dla liczby klas równej *9*, wraz ze wzrostem liczby klas w podzbiorze, dokładność rosła. Za wyjątkiem wartości *10*, gdyż dla klasyfikatora *knn* zanotowano tutaj drobny spadek dokładności.

Zachowanie klasyfikacji dla zbioru [3] wygląda podobnie jak dla zbioru [1], tzn. można zaobserwować powolny wzrost dokładności, dla obu klasyfikatorów tendencja jest wzrostowa, z pojedynczymi zawahaniami.

4. Wnioski

Podsumowując wykonane zadanie wnioskujemy, że:

— W metodzie *PCA* przy wybranych zbiorach zmiana *svd_solver* nie wpływała znacząco na wyniki
— W metodzie *PCA* najlepsze wyniki uzyskiwano dla najmniejszych wartości *n_components*
— W metodzie *SVD* przy wybranych zbiorach danych zmiana algorytmu nie wpływała znacząco na wyniki
— W metodzie *SVD* najlepsze wyniki uzyskiwano dla najmniejszych wartości *n_components*
— Trudno znaleźć szybko działającą implementację metody *CFS*
— Duży wpływ na działanie metody *CFS* ma wartość korelacji pomiędzy klasami zbioru.
— W metodzie *CFS* można zaobserwować, że nie wszystkie wyznaczone przez algorytm podzbiory są lepsze od pozostałych, tzn. nie można z całkowitą pewnością powiedzieć, że im więcej elementów ma podzbiór klas, tym na jego podstawie klasyfikacja będzie dokładniejsza.
— Czasami udaje się uzyskać wyższą jakość klasyfikacji na zredukowanym zbiorze danych, niż na oryginalnym
— Często zdarza się, że większość cech w zbiorze danych jest nadmiarowa
— Czym bardziej abstrakcyjne i złożone dane reprezentuje zbiór, tym więcej cech wymaga taki zbiór do poprawnej klasyfikacji.

— Niektóre klasyfikatory (jak las losowy) radzą sobie lepiej ze zredukowanymi zbiorami danych, a inne (np. KNN), są bardziej podatne na pojedyncze braki w wartościach cech próbek uczących

Literatura

[1] https://archive.ics.uci.edu/ml/datasets/isolet
[2] https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[3] https://archive.ics.uci.edu/ml/datasets/CNAE-9