

Forest Cover Type Prediction - dokumentacja (K11)

Zaawansowane uczenie maszynowe

Paweł Młyniec
Hubert Borkowski

28 lutego 2021

Spis treści

1	Wstęp	2
2	Interpretacja tematu	2
2.1	Opis danych wejściowych	2
2.2	Opis atrybutu dyskretnego reprezentującego pojęcie docelowe	2
3	Wstępne obrobienie danych	2
4	Opis algorytmów klasyfikacji	2
4.1	Klasyfikator bayesowski	2
4.2	Algorytm K najbliższych sąsiadów	3
4.3	Drzewo decyzyjne	3
5	Przebieg badań	3
6	Wyniki	4
6.1	Analiza zbiorów danych	4
6.1.1	Liczebność klas	4
6.1.2	Współczynnik skośności	4
6.1.3	Korelacja pomiędzy atrybutami	4
6.1.4	Wykresy pudełkowe	5
6.1.5	Selekcja atrybutów algorytmem lasso	9
6.2	Zdefiniowanie nowych atrybutów algorytmami PCA i MCA	10
6.3	Wyniki testowania algorytmów	10
6.4	10-krotna walidacja krzyżowa z 3 powtórzeniami z selekcją parametrów LASSO	10
6.4.1	Naiwny klasyfikator bayesowski	10
6.4.2	Algorytm K najbliższych sąsiadów	10
6.4.3	Drzewo decyzyjne	11
6.4.4	eXtreme Gradient Boosting Tree	11
6.5	10-krotna walidacja krzyżowa z 3 powtórzeniami bez selekcji parametrów	12
6.5.1	Naiwny klasyfikator bayesowski	12
6.5.2	Algorytm K najbliższych sąsiadów	12
6.5.3	Drzewo decyzyjne	13
6.5.4	eXtreme Gradient Boosting Tree	13
6.6	10-krotna walidacja krzyżowa z 10 powtórzeniami z selekcją parametrów LASSO	13
6.6.1	Naiwny klasyfikator bayesowski	13
6.6.2	Algorytm K najbliższych sąsiadów	14
6.6.3	Drzewo decyzyjne	14
6.6.4	eXtreme Gradient Boosting Tree	15
6.7	10-krotna walidacja krzyżowa z 10 powtórzeniami bez selekcji parametrów	15
6.7.1	Naiwny klasyfikator bayesowski	15
6.7.2	Algorytm K najbliższych sąsiadów	16
6.7.3	Drzewo decyzyjne	16
6.7.4	eXtreme Gradient Boosting Tree	17
6.8	Podsumowanie wyników	17
7	Bibliografia	19

1 Wstęp

Celem projektu jest klasyfikacja rodzaju lasu na podstawie zmiennych kartograficznych. Dane oraz ich opis jest przedstawiony w [1].

2 Interpretacja tematu

2.1 Opis danych wejściowych

W projekcie będzie analizowany zbiór danych zawierający dane o typie lasu rosnącego na danym obszarze parku narodowego im. Roosevelta w Kolorado w USA. Zbiór składa się z 15 120 obserwacji w zbiorze treningowym oraz 565 892 obserwacji w zbiorze testowym.

Każda obserwacja dotyczy obszaru na kwadracie o wymiarach 30m na 30m. Zawiera 118 cech takich jak np. typ gleby, nasłonecznienie. Dokładny opis danych znajduje się na stronie [2].

2.2 Opis atrybutu dyskretnego reprezentującego pojęcie docelowe

Z podanych w zbiorze 7 typów drzew.

- 1 - Jodłowo-świerkowe
- 2 - Sosna wydymowa
- 3 - Sosna żółta
- 4 - Topola
- 5 - Osika
- 6 - Dąb zielony
- 7 - Krzywulec

Przedmiotem klasyfikacji w tym zadaniu będzie predykcja na podstawie parametrów terenu, który gatunek dominuje na danym obszarze.

3 Wstępne obrobienie danych

Na początku zostanie przeprowadzona wstępna analiza danych pod kątem wartości brakujących. W przypadku małej ilości danych brakujących takie obserwacje zostaną usunięte. W przeciwnym wypadku dane zostaną uzupełnione odpowiednio dobranym sposobem.

Dane zawierające wartości liczbowe (kolumny: Elevation, Aspect, Slope, Horizontal_Distance_To_Hydrology, Vertical_Distance_To_Hydrology, Horizontal_Distance_To_Roadways, Hillshade_9am, Hillshade_Noon, Hillshade_3pm, Horizontal_Distance_To_Fire_Points) zostaną podjęte standaryzacji.

Następnie zostanie zbadana możliwość wyboru nowych parametrów algorytmem PCA [3] z kryterium części wyjaśnionej wariancji o wartości 95%.

Selekcja atrybutów będzie dokonana algorytmem regresji LASSO.

Jeśli analiza wstępna wykaże nierówne rozłożenie klas można także zrobić undersampling rzadziej występującej klasy.

4 Opis algorytmów klasyfikacji

4.1 Klasyfikator bayesowski

Naiwny klasyfikator Bayes’a jest statystycznym klasyfikatorem, opartym na twierdzeniu Bayes’a. Zastosowanie wzoru Bayes’a do zadania klasyfikacji może polegać na wyznaczeniu prawdopodobieństwa pewnej

klasy d dla pewnego przykładu x , czyli $P(d|x)$. Naiwny klasyfikator Bayes’a zakłada, że wartości atrybutów w klasach są niezależne. Założenie to jest zwane założeniem o niezależności warunkowej klasy (ang. class conditional independence).

4.2 Algorytm K najbliższych sąsiadów

Algorytm k najbliższych sąsiadów próbuje przyporządkować punkt do klasy na podstawie wybranej liczby k sąsiednich punktów. Głównym założeniem tego algorytmu jest to, że punkty leżące blisko siebie w przestrzeni, a tym samym punkty o zbliżonych do siebie cechach, powinny należeć do tej samej klasy. To dość proste założenie jest spełnione w wielu przypadkach praktycznego zastosowania takiego algorytmu. W związku z tym algorytm ten potrafi dać bardzo dobre wyniki w wielu zastosowaniach, pomimo swojej koncepcyjnej prostoty.

4.3 Drzewo decyzyjne

Klasyfikator drzewa decyzyjnego (ang. Decision Tree Classifier) na bazie danych wejściowych podzielonych na klasy tworzy strukturę drzewiastą, na podstawie której można potem przydzielić nieznany wcześniej punkt danych do którejś ze zdefiniowanych klas. W węzłach powstałego w ten sposób drzewa znajdują się informacje pozwalające skierować badany punkt na konkretną ścieżkę w dół drzewa na podstawie jego cech. Natomiast w liściach drzewa znajdują się wartości prawdopodobieństwa z jakimi punkt danych, który dotarł do danego liścia, może przynależeć do odpowiedniej klasy.

5 Przebieg badań

- Dane zostały pobrane i wczytane do formatu DataFrame.
- Przeprowadzona została analiza zbioru danych. W tym:
 - Liczności klas dla których będzie przeprowadzona klasyfikacja
 - Współczynnika skośności
 - Korelacji pomiędzy cechami obserwacji
 - Wyliczona została heatmapa kowariancji atrybutów
 - Przedstawione wykresy pudełkowe dla każdej z atrybutów
- Selekcja atrybutów algorytmem lasso
- Próba zdefiniowania nowych atrybutów. Dla wartości numerycznych algorytmem PCA, a dla kategori-
cznych algorytmem MCA
- Po testach została odrzucona koncepcja zakładająca wykorzystanie algorytmu PCA do wyboru para-
metrów jako nie dająca wystarczających korzyści dla uczenia. Podobnie odrzucony został klasyfikator
SVM jako nie dość dostosowany do problemu klasyfikacji wielklasowej.
- Zdecydowano się na zmianę zbioru testowanych modeli na: naiwny klasyfikator bayesowski, algorytm
 k najbliższych sąsiadów, drzewa decyzyjne oraz eXtreme Gradient Boosting tree.
- Dla każdego klasyfikatora przeprowadzono test zawierający uczenie z doбором parametrów przy użyciu
10-krotnej walidacji krzyżowej z 3 i 10 powtórzeniami z selekcją parametrów przy użyciu algorytmu
LASSO i bez selekcji. Dla każdego modelu został zmierzony czas treningu i czas predykcji dokonywanej
na zbiorze walidacyjnym.
- Badania zostały wykonane przy włączeniu opcji przetwarzania współbieżnego z pomocą pakietu języka
R *doParallel* w rozbiciu na 7 wątków, na maszynie z procesorem Intel i7-4770K z 8 rdzeniami o
taktowaniu 3.7GHz i 16GB pamięci RAM.

6 Wyniki

6.1 Analiza zbiorów danych

6.1.1 Liczebność klas

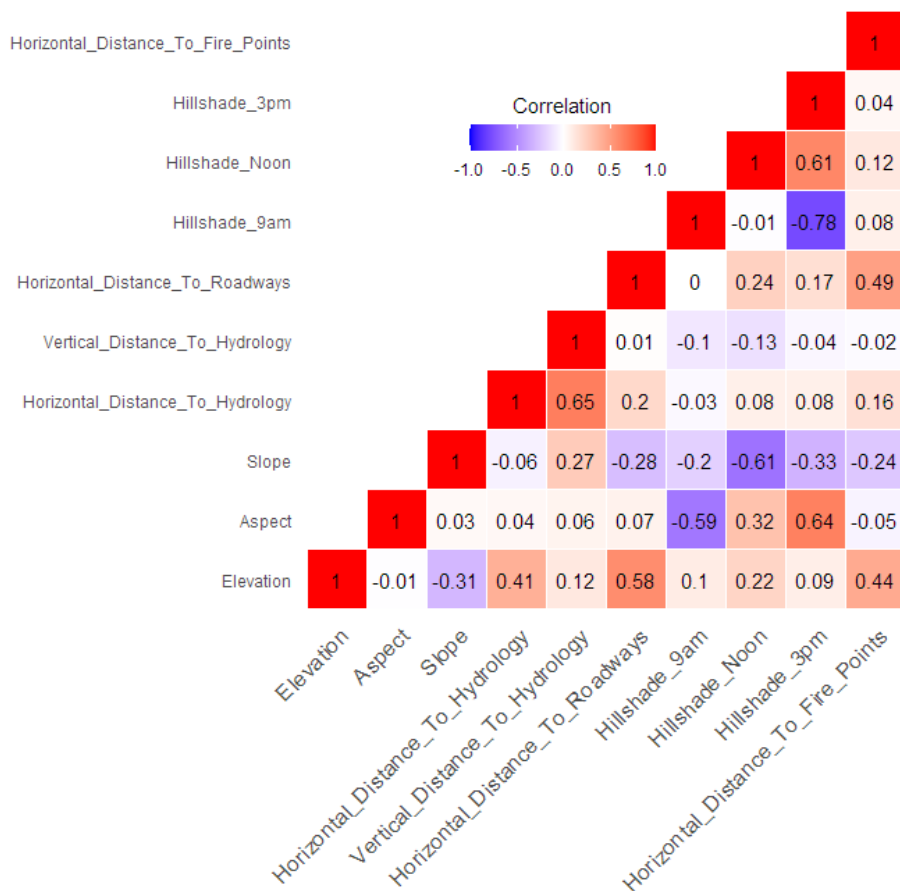
Wszystkie klasy są równoliczne i posiadają po 2160 obserwacji. Nie ma więc celu próba równoważenia liczebności klas.

6.1.2 Współczynnik skośności

W celu zbadania jak bardzo rozkład różnił się od normalnego został wyliczony współczynnik skośności. Dwa rodzaje gleby mają rozkład mocno przesunięty w kierunku wyższych wartości. Są to atrybuty *Soil_Type8* oraz *Soil_Type25* z współczynnikami skośności równymi 122.95.

Kolejne dwa atrybuty mają natomiast wartości wyliczone, które nie są liczbami. Są to atrybuty *Soil_Type7* oraz *Soil_Type15*. Wnika to z faktu, że jako kategoria nie występują w zbiorze danych, więc w celu poprawnego liczenia kolejnych algorytmów nie będą uwzględnione.

6.1.3 Korelacja pomiędzy atrybutami

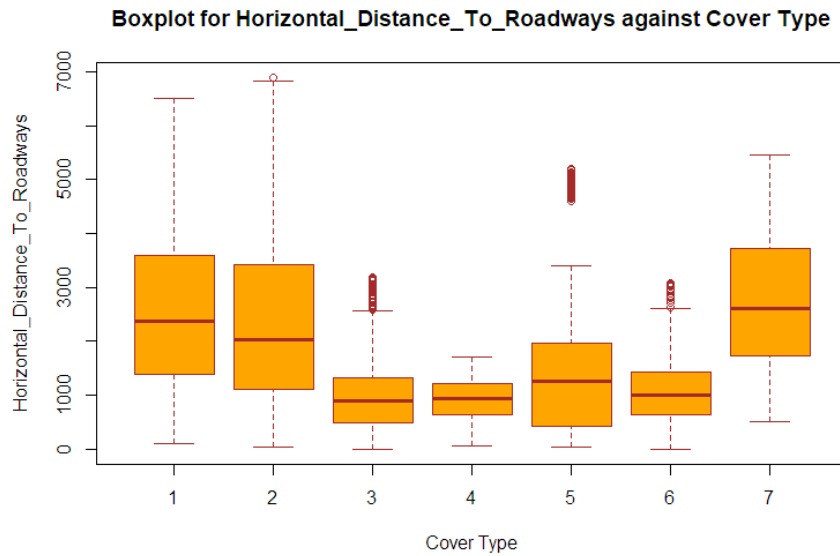


Rysunek 1: Heatmapa kowariancji pomiędzy atrybutami numerycznymi

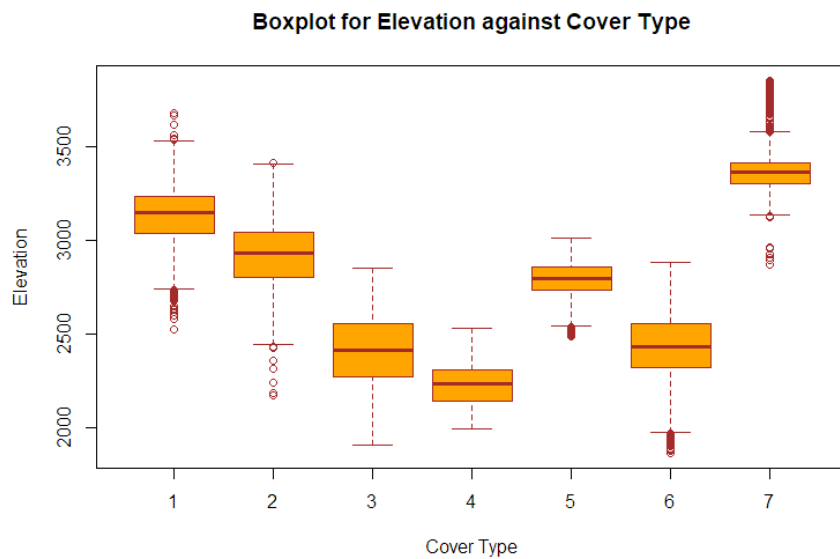
Poza cieniami o różnych porach roku pozostałe atrybuty są raczej nieskorelowane. W niskiej korelacji wynika także, że raczej bezcelowa jest próba zastąpienia atrybutów mniejszą ilością algorytmem PCA.

6.1.4 Wykresy pudełkowe

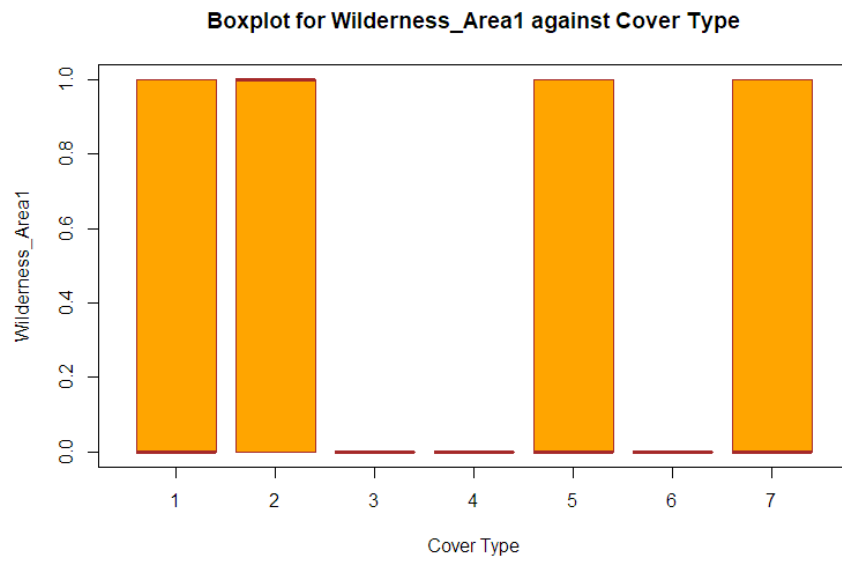
Wykresy pudełkowe zostały sporządzone dla każdego atrybutu w zależności od rodzaju lasu. Z powodu ich znaczącej ilości zostały zamieszczone tylko niektóre z nich.



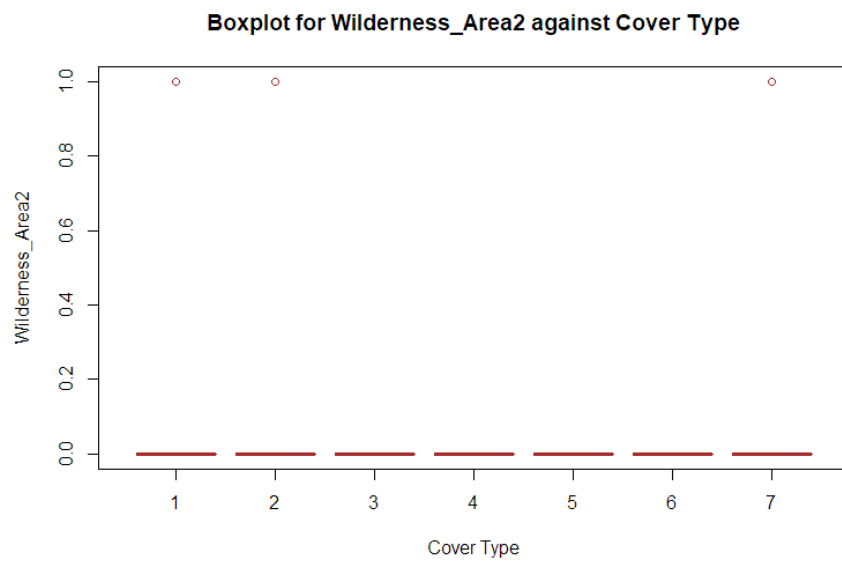
Rysunek 2: Wykres pudełkowy dla odległości od drogi dla rodzajów lasu



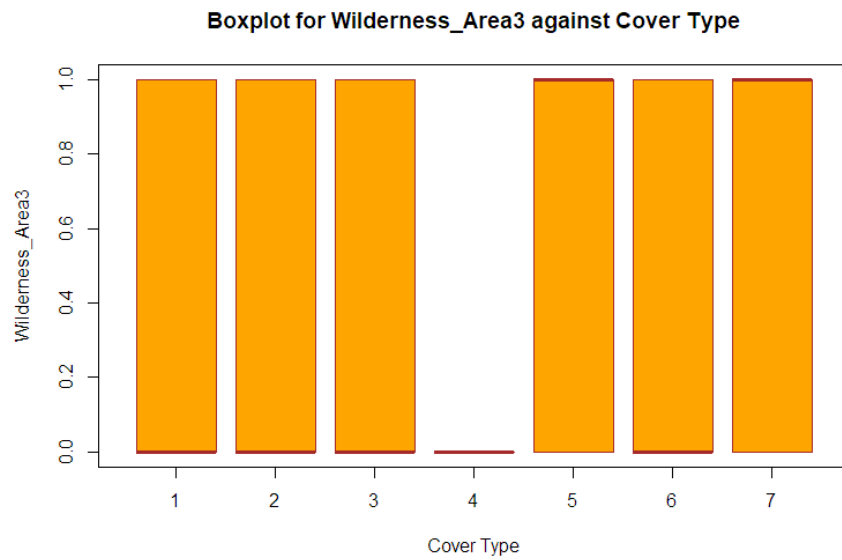
Rysunek 3: Wykres pudełkowy dla nachylenia dla rodzajów lasu



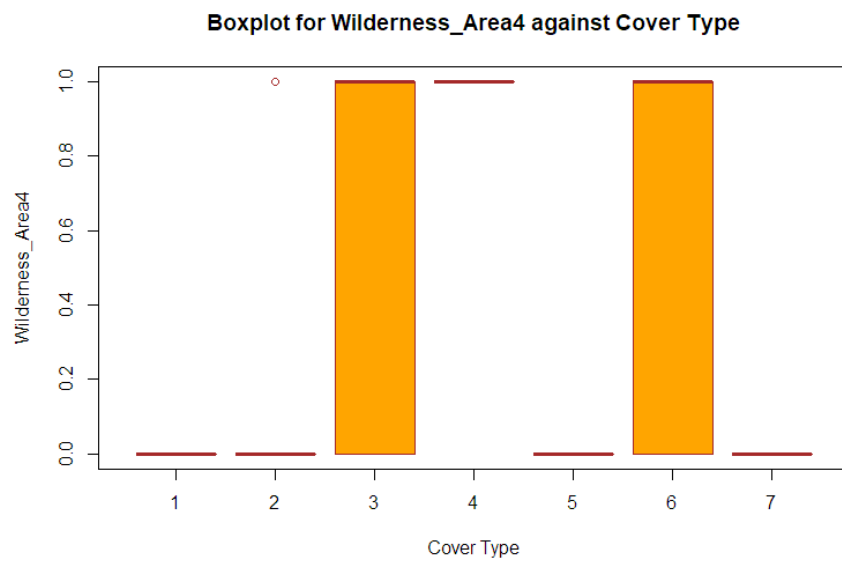
Rysunek 4: Wykres pudełkowy dla obszaru Rawah dla rodzajów lasu



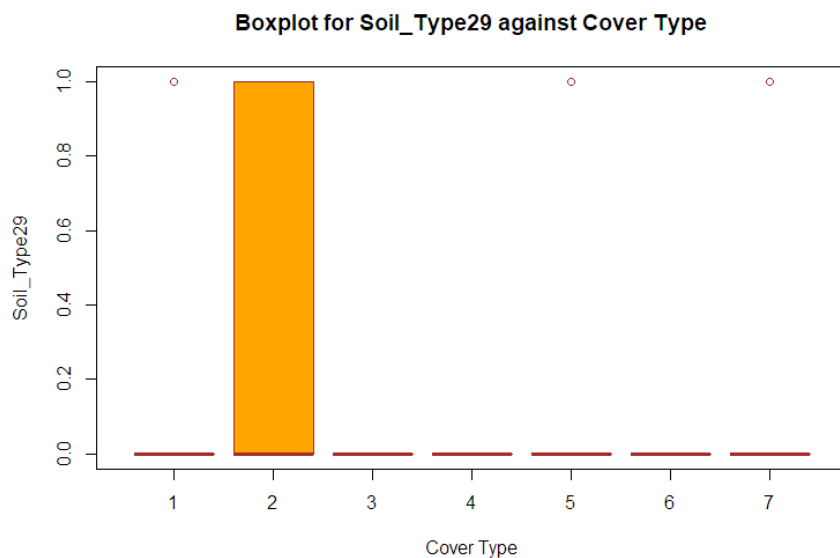
Rysunek 5: Wykres pudełkowy dla obszaru Neota dla rodzajów lasu



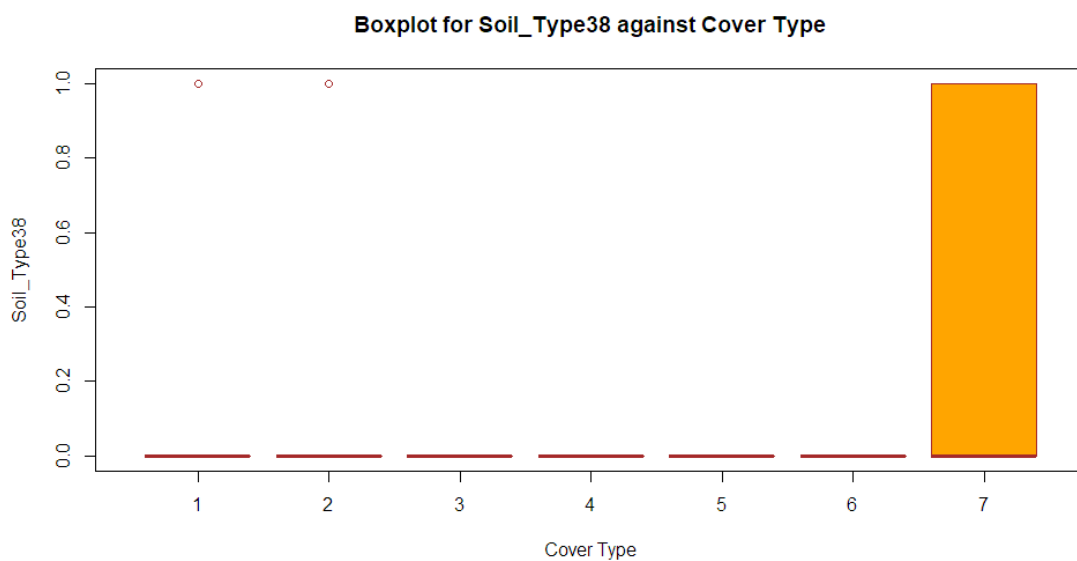
Rysunek 6: Wykres pudełkowy dla obszaru Comanche Peak dla rodzajów lasu



Rysunek 7: Wykres pudełkowy dla obszaru Comanche Peak dla rodzajów lasu



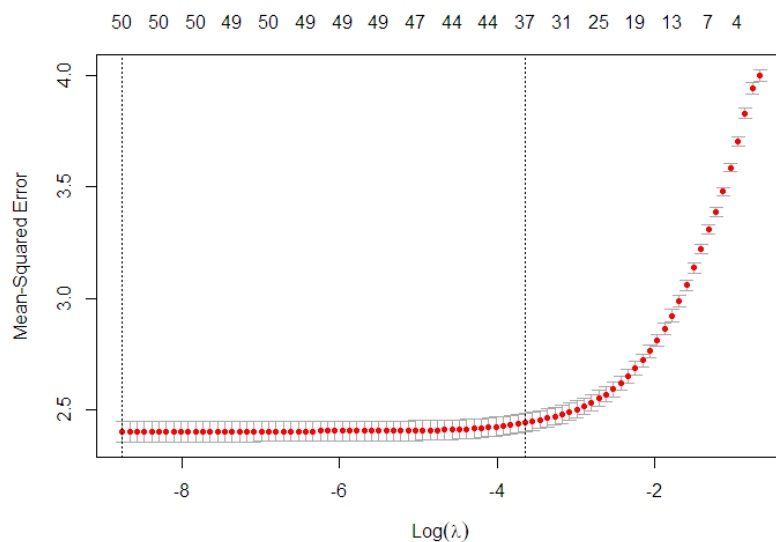
Rysunek 8: Wykres pudełkowy dla typu gleby Como dla rodzajów lasu



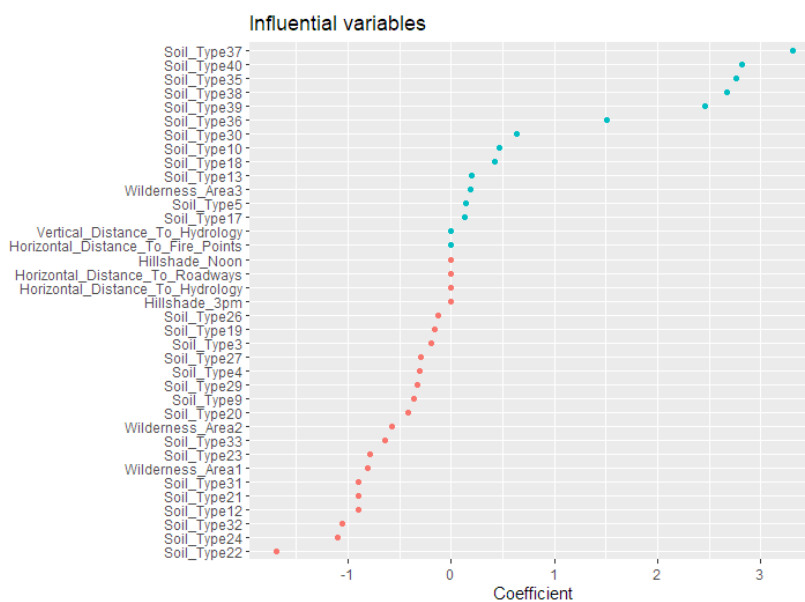
Rysunek 9: Wykres pudełkowy dla typu gleby Moran dla rodzajów lasu

Z tych wykresów wynika, że Topola rośnie raczej bliżej dróg, w obszarach o niskim nachyleniu. Krzywulec natomiast rośnie na zboczach o wysokim nachyleniu dalej od dróg. W obszarze Rawah rosną raczej jodły, świerki, sosny wydymowe, osiki oraz krzywulce. W obszarze Comanche Peak Sosny żółte, topole i daglezie. W obszarze Como sosny wyspowe, a w Comanche Peak wszystkie rodzaje poza topolą. Na glebie Como rosną głównie sosny wydymowe, a na glebie Moran krzywulce.

6.1.5 Selekcja atrybutów algorytmem lasso



Rysunek 10: Wykres straty średnio kwadratowej w zależności od logarytmu parametru lambda dla regresji lasso



Rysunek 11: Wykres istotności zmiennych

Z powyższych wykresów wynika, że wraz z rosnącym parametrem lambda odpowiadającym pośrednio za selekcję zmiennych, aż do uzyskania 37 atrybutów nie zmienia się błąd regresji. Niestety przy rosnącym logarytmie z lambda na wykresie nie poprawia się też wartość błędu nie można więc stwierdzić, że selekcja atrybutów poprawi predykcję w przyszłości. W celu zbadania wpływu selekcji na wynik wybrano zmienne przedstawione na drugim wykresie i przeprowadzono dla nich dalsze testy.

6.2 Zdefiniowanie nowych atrybutów algorytmami PCA i MCA

Z przeprowadzonych wcześniej badań korelacji pomiędzy zmiennymi wynika, że nie ma sensu stwarzania nowych zmiennych numerycznych algorytmem PCA.

Z racji posiadania tylko dwóch zmiennych kategoriycznych stworzenie nowych zmiennych algorytmem MCA też mija się z celem.

6.3 Wyniki testowania algorytmów

Wymienione wcześniej algorytmu przetestowano trenując je przy użyciu 10-krotnej walidacji krzyżowej z powtórzeniami. Dobór parametrów poszczególnych algorytmów odbywał się automatycznie. Walidacja krzyżowa była ustawiona na minimalizowanie parametru *logloss* zgodnie z zaleceniem dołączonym do zbioru danych na stronie Kaggle.

Dodatkowo przed treningiem dane zostały wyskalowane z parametrami (*center*, *scale*).

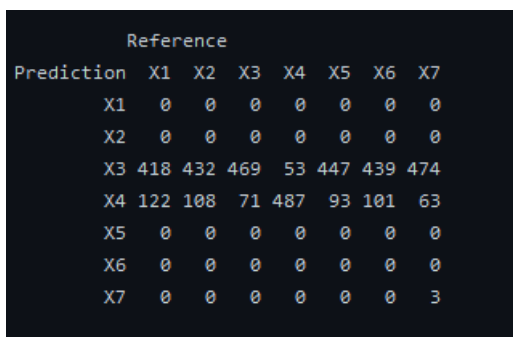
W celu porównania wpływu wyboru parametrów metodą LASSO na czas i efektywność szkolenia testy zostały przeprowadzone dla wariantu z wyborem LASSO i bez takiego wyboru parametrów.

Poniżej widoczne są uzyskane w testach wyniki.

6.4 10-krotna walidacja krzyżowa z 3 powtórzeniami z selekcją parametrów LASSO

6.4.1 Naiwny klasyfikator bayesowski

- Czas uczenia: 15.25 s
- Czas predykcji: 15.41 s
- AUC: 0.77
- precyzja: 0.25
- Wybrane parametry: *laplace* = 0, *usekernel* = TRUE and *adjust* = 1



	Reference						
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	0	0	0	0	0	0	0
X2	0	0	0	0	0	0	0
X3	418	432	469	53	447	439	474
X4	122	108	71	487	93	101	63
X5	0	0	0	0	0	0	0
X6	0	0	0	0	0	0	0
X7	0	0	0	0	0	0	3

6.4.2 Algorytm K najbliższych sąsiadów

- Czas uczenia: 2.02 min
- Czas predykcji: 1.017 s
- AUC: 0.93
- precyzja: 0.79
- Wybrane parametry: *kmax* = 9, *distance* = 2 and *kernel* = optimal

	Reference						
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	355	105	0	0	3	1	12
X2	106	316	6	0	8	14	1
X3	3	17	383	11	9	65	0
X4	0	1	40	511	0	31	0
X5	34	59	18	0	512	14	3
X6	9	24	93	18	6	415	0
X7	33	18	0	0	2	0	524

6.4.3 Drzewo decyzyjne

- Czas uczenia: 18.12 s
- Czas predykcji: 0.51 s
- AUC: 0.93
- precyzja: 0.76

	Reference						
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	340	136	0	0	13	6	19
X2	130	302	11	0	33	20	6
X3	5	19	396	28	19	108	0
X4	0	1	33	496	0	31	0
X5	29	49	12	0	472	16	0
X6	5	24	88	16	3	359	0
X7	31	9	0	0	0	0	515

6.4.4 eXtreme Gradient Boosting Tree

- Czas uczenia: 24.39 min
- Czas predykcji: 0.08 s
- AUC: 0.96
- precyzja: 0.76
- Wybrane parametry: nrounds = 150, max_depth = 3, eta = 0.4, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample = 0.75

	Reference						
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	346	124	0	0	18	2	23
X2	127	295	5	0	27	15	12
X3	1	20	384	23	26	90	1
X4	0	0	27	507	0	24	0
X5	22	61	27	0	456	13	2
X6	8	26	97	10	13	396	0
X7	36	14	0	0	0	0	502

6.5 10-krotna walidacja krzyżowa z 3 powtórzeniami bez selekcji parametrów

6.5.1 Naiwny klasyfikator bayesowski

- Czas uczenia: 16.63 s
- Czas predykcji: 16.80 s
- AUC: 0.82
- precyzja: 0.32
- Wybrane parametry: laplace = 0, usekernel = TRUE and adjust = 1

Reference							
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	15	20	0	0	0	0	0
X2	0	0	0	0	0	0	0
X3	445	497	500	91	537	510	313
X4	0	1	40	449	0	30	0
X5	4	4	0	0	3	0	0
X6	0	0	0	0	0	0	0
X7	76	18	0	0	0	0	227

6.5.2 Algorytm K najbliższych sąsiadów

- Czas uczenia: 2.68 min
- Czas predykcji: 1.12 s
- AUC: 0.89
- precyzja: 0.81
- Wybrane parametry: kmax = 9, distance = 2 and kernel = optimal

Reference							
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	376	89	0	0	5	1	19
X2	107	359	12	0	17	15	0
X3	1	11	389	14	6	75	0
X4	0	0	32	508	0	29	0
X5	23	55	18	0	509	10	1
X6	4	16	89	18	3	410	0
X7	29	10	0	0	0	0	520

6.5.3 Drzewo decyzyjne

- Czas uczenia: 24.19 s
- Czas predykcji: 0.69 s
- AUC: 0.93
- precyzja: 0.79

Reference							
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	359	116	0	0	5	0	30
X2	123	329	6	0	30	22	2
X3	0	12	412	18	8	72	0
X4	0	0	29	503	0	18	0
X5	10	56	7	0	488	11	1
X6	3	16	86	19	9	417	0
X7	45	11	0	0	0	0	507

6.5.4 eXtreme Gradient Boosting Tree

- Czas uczenia: 30.72 min
- Czas predykcji: 0.09 s
- AUC: 0.97
- precyzja: 0.81
- Wybrane parametry: nrounds = 150, max_depth = 3, eta = 0.4, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample = 0.75

Reference							
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	399	120	0	0	1	0	21
X2	81	328	5	0	22	7	0
X3	1	12	422	16	11	65	0
X4	0	0	22	519	0	12	0
X5	12	58	9	0	505	7	0
X6	1	14	82	5	1	449	0
X7	46	8	0	0	0	0	519

6.6 10-krotna walidacja krzyżowa z 10 powtórzeniami z selekcją parametrów LASSO

6.6.1 Naiwny klasyfikator bayesowski

- Czas uczenia: 1.29 min

- Czas predykcji: 1.29 min
- AUC: 0.77
- precyzja: 0.25
- Wybrane parametry: laplace = 0, usekernel = TRUE and adjust = 1

Reference							
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	0	0	0	0	0	0	0
X2	0	0	0	0	0	0	0
X3	418	432	469	53	447	439	474
X4	122	108	71	487	93	101	63
X5	0	0	0	0	0	0	0
X6	0	0	0	0	0	0	0
X7	0	0	0	0	0	0	3

6.6.2 Algorytm K najbliższych sąsiadów

- Czas uczenia: 7.60 min
- Czas predykcji: 0.99 s
- AUC: 0.93
- precyzja: 0.79
- Wybrane parametry: kmax = 9, distance = 2 and kernel = optimal

Reference							
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	355	105	0	0	3	1	12
X2	106	316	6	0	8	14	1
X3	3	17	383	11	9	65	0
X4	0	1	40	511	0	31	0
X5	34	59	18	0	512	14	3
X6	9	24	93	18	6	415	0
X7	33	18	0	0	2	0	524

6.6.3 Drzewo decyzyjne

- Czas uczenia: 49.90 s
- Czas predykcji: 0.53 s
- AUC: 0.93
- precyzja: 0.76

	Reference						
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	340	136	0	0	13	6	19
X2	130	302	11	0	33	20	6
X3	5	19	396	28	19	108	0
X4	0	1	33	496	0	31	0
X5	29	49	12	0	472	16	0
X6	5	24	88	16	3	359	0
X7	31	9	0	0	0	0	515

6.6.4 eXtreme Gradient Boosting Tree

- Czas uczenia: 1.38 h
- Czas predykcji: 0.07 s
- AUC: 0.96
- precyzja: 0.76
- Wybrane parametry: nrounds = 150, max_depth = 3, eta = 0.4, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample = 0.75

	Reference						
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	334	131	0	0	9	5	21
X2	141	283	5	0	24	9	14
X3	2	19	384	20	20	94	0
X4	0	0	29	508	0	21	0
X5	22	60	22	0	475	14	1
X6	7	32	100	12	12	397	0
X7	34	15	0	0	0	0	504

6.7 10-krotna walidacja krzyżowa z 10 powtórzeniami bez selekcji parametrów

6.7.1 Naiwny klasyfikator bayesowski

- Czas uczenia: 49.66 s
- Czas predykcji: 49.83 s
- AUC: 0.82
- precyzja: 0.32
- Wybrane parametry: laplace = 0, usekernel = TRUE and adjust = 1

	Reference						
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	18	23	0	0	0	0	0
X2	0	0	0	0	0	0	0
X3	444	491	505	76	536	514	322
X4	4	0	35	464	0	26	1
X5	2	8	0	0	4	0	0
X6	0	0	0	0	0	0	0
X7	72	18	0	0	0	0	217

6.7.2 Algorytm K najbliższych sąsiadów

- Czas uczenia: 8.46 min
- Czas predykcji: 1.43 s
- AUC: 0.92
- precyzja: 0.80
- Wybrane parametry: kmax = 9, distance = 2 and kernel = optimal

	Reference						
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	368	108	0	0	4	1	14
X2	95	327	9	0	11	4	9
X3	1	13	390	23	10	73	0
X4	0	1	36	505	0	22	0
X5	25	69	10	0	502	6	2
X6	5	21	95	12	13	434	0
X7	46	1	0	0	0	0	515

6.7.3 Drzewo decyzyjne

- Czas uczenia: 1.12 min
- Czas predykcji: 0.68 s
- AUC: 0.93
- precyzja: 0.79

Reference							
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	351	120	0	0	10	1	25
X2	126	320	11	0	30	5	14
X3	0	20	411	35	10	84	0
X4	0	0	23	488	0	6	0
X5	11	60	12	0	483	8	0
X6	2	17	83	17	7	436	0
X7	50	3	0	0	0	0	501

6.7.4 eXtreme Gradient Boosting Tree

- Czas uczenia: 1.70 h
- Czas predykcji: 0.09 s
- AUC: 0.97
- precyzja: 0.83
- Wybrane parametry: nrounds = 150, max_depth = 3, eta = 0.4, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample = 0.75

Reference							
Prediction	X1	X2	X3	X4	X5	X6	X7
X1	383	104	0	0	2	0	22
X2	97	346	4	0	20	3	3
X3	0	13	437	16	10	78	0
X4	0	0	31	513	0	11	0
X5	10	61	8	0	501	1	0
X6	2	15	60	11	7	447	0
X7	48	1	0	0	0	0	515

6.8 Podsumowanie wyników

Przyglądając się uzyskanym wynikom można zauważyć, że zastosowanie walidacji krzyżowej z 10 powtórzeniami nie przynosi istotnej poprawy w jakości modelu w porównaniu do wersji z 3 powtórzeniami, za to znacząco zwiększa czas treningu klasyfikatorów.

Jeśli chodzi o zastosowanie selekcji parametrów przy użyciu algorytmu LASSO, to widać, że skraca ono proces uczenia klasyfikatorów, a także w niewielkim stopniu pozwala na poprawę jakości predykcji.

Analizując wyniki poszczególnych klasyfikatorów można natomiast dojść do następujących wniosków:

- *Naiwny klasyfikator bayesowski* - najkrótszy czas treningu i zarazem najniższa skuteczność, analizując tabelę pomyłek widać, że ten model nie radzi sobie dobrze z klasyfikacją wieloklasową dla tych danych;
- *Algorytm K najbliższych sąsiadów (knn)* - widać znacznie lepsze wyniki niż w przypadku poprzedniego algorytmu przy nadal małym czasie uczenia, najgorzej poradził sobie z klasami 1, 2 i 3;

- *Drzewo decyzyjne* - czas uczenia podobny do pierwszego klasyfikatora, natomiast dużo lepsze efekty, osiąga wyniki zbliżone do K najbliższych sąsiadów;
- *eXtreme Gradient Boosting Tree* - zdecydowanie najbardziej czasochłonny pod względem uczenia modelu, natomiast najkrótszy czas predykcji, mimo dużo dłuższego uczenia nie radzi sobie z predykcją znacząco lepiej niż poprzednio omawiane algorytmy: drzewo decyzyjne i knn

Można również zauważyć, że w wyniku walidacji krzyżowej z różnymi ustawieniami i selekcją kolumn dobierane zostały te same parametry wejściowe algorytmów.

Pełne logi z wykonywanych testów można zobaczyć w plikach dołączonych do pliku z kodem źródłowym programu testującego modele.

7 Bibliografia

- [1] *Forest Cover Type Prediction*, <https://www.kaggle.com/c/forest-cover-type-prediction/overview>
- [2] *Forest Cover Type Prediction, data*, <https://www.kaggle.com/c/forest-cover-type-prediction/data>
- [3] Pearson K. *On lines and planes of closest fit to systems of points in space* Philos Mag A. 6: 559-572.